



TUGAS AKHIR TE 141599

PENGENDALI KECEPATAN PADA ALAT SENTRIFUGASI MENGUNAKAN METODE LOGIKA FUZZY

Garudio Kusuma Aji
NRP 07111040000172

Dosen Pembimbing
Dr. Ir. Djoko Purwanto, M. Eng.
Dr. Muhammad Rivai, ST., MT.

Departemen Teknik Elektro
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2018



TUGAS AKHIR TE 141599

**PENGENDALI KECEPATAN PADA ALAT SENTRIFUGASI
MENGUNAKAN METODE LOGIKA FUZZY**

Garudio Kusuma Aji
NRP 07111040000172

Dosen Pembimbing
Dr. Ir. Djoko Purwanto, M. Eng.
Dr. Muhammad Rivai, ST., MT.

Departemen Teknik Elektro
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2018



FINAL PROJECT TE 141599

***SPEED CONTROLLER ON CENTRIFUGATION TOOLS USING
FUZZY LOGIC METHOD***

Garudio Kusuma Aji
NRP 07111040000172

Supervisor
Dr. Ir. Djoko Purwanto, M. Eng.
Dr. Muhammad Rivai, ST., MT.

Electrical Engineering Departement
Faculty of Electrical Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2018

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul “PENGENDALI KECEPATAN PADA ALAT SENTRIFUGASI MENGGUNAKAN METODE LOGIKA FUZZY” adalah benar-benar hasil karya intelektual sendiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya orang lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai dengan peraturan yang berlaku.

Surabaya, 20 Juli 2018



Garudjo Kusuma Aji
NRP. 07111040000172

.....*Halaman ini sengaja dikosongkan*.....

**PENGENDALI KECEPATAN PADA
ALAT SENTRIFUGASI
MENGUNAKAN METODE LOGIKA FUZZY**

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik
Pada
Bidang Studi Elektronika
Departemen Teknik Elektro
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember**

Menyetujui :

Dosen Pembimbing 1



Dr. Ir. Djoko Purwanto, M. Eng.
NIP. 196512111990021002

Dosen Pembimbing 2



Dr. Muhammad Rivai, ST., MT.
NIP. 196904261994031003



.....*Halaman ini sengaja dikosongkan*.....

Pengendali Kecepatan pada Alat Sentrifugasi Menggunakan Metode Logika Fuzzy

Nama : Garudio Kusuma Aji
Pembimbing 1 : Dr. Ir. Djoko Purwanto, M. Eng.
Pembimbing 2 : Dr. Muhammad Rivai, ST., MT.

ABSTRAK

Keandalan alat sentrifugasi yang banyak digunakan para pengusaha *Virgin Coconut Oil* (VCO) saat ini masih terbilang rendah. Alat sentrifugasi yang ada saat ini cenderung diperuntukkan dalam proses pembuatan VCO dengan pemerasan santan tanpa penambahan air bukan untuk pemerasan santan dengan penambahan air. Pada penelitian ini telah dirancang dan dibuat sistem pengendali kecepatan pada alat sentrifugasi dengan metode logika fuzzy untuk meningkatkan keandalan alat sentrifugasi dalam proses pembuatan VCO, baik menggunakan santan murni atau santan yang sudah tercampur air. Logika fuzzy akan menentukan *set point* kecepatan berdasarkan masukan berupa konsentrasi atau kekentalan santan dan waktu proses. *Set point* kecepatan sentrifugasi keluaran logika fuzzy akan menjadi nilai parameter oleh sistem kestabilan kecepatan. Sistem pengendali kecepatan pada alat sentrifugasi mampu mencapai *settling time* pada detik ke-33 dengan *set point* 400 dan 800 rpm pada keadaan tanpa beban, sistem mencapai *settling time* pada detik ke-35 dengan *set point* 600 rpm pada keadaan berbeban. Sistem pengendali kecepatan berhasil meningkatkan keandalan alat sentrifugasi, terbukti dalam durasi 5 menit berhasil memisahkan santan dengan tingkat konsentrasi berbedadengan rata-rata keberhasilan 87,24%.

Kata Kunci: alat sentrifugasi, konsentrasi, logika fuzzy, waktu proses

.....*Halaman ini sengaja dikosongkan*.....

Speed Controller on Centrifugation Tools Using Fuzzy Logic Method

Name : Garudio Kusuma Aji
Supervisor 1 : Dr. Ir. Djoko Purwanto, M. Eng.
Supervisor 2 : Dr. Muhammad Rivai, ST., MT.

ABSTRACT

The reliability of centrifugation tools used by Virgin Coconut Oil (VCO) entrepreneurs is still relatively low. The present centrifugation tools tends to be used in the process of making VCO by squeezing the coconut milk without adding water, not to the coconut milk from squeezing with addition of water. In this research has been designed and made speed controller system on centrifugation machine using fuzzy logic method, to increase the reliability of centrifugation machine in the process of making VCO, either using pure coconut milk or coconut milk that has been mixed with water. Fuzzy logic will determine set point of the speed based on the input of the concentration or the thickness of the coconut milk and the process time. The set point of centrifugation speed of fuzzy logic output become parameter numbers of the speed stability system. The speed control system on the centrifugation tools can reach the settling time at 33 seconds with set point 400 and 800 rpm at no-load condition, the system reaches the settling time at 35 seconds with a set point of 600 rpm at loaded condition. The speed control system succeeds in improving the reliability of centrifugation tools, proven within 5 minutes of successfully separating the coconut milk with different concentration levels with averages succeeds 87,24%.

Keywords: Centrifugation Tools, Concentration, Fuzzy Logic, Process time.

.....*Halaman ini sengaja dikosongkan*.....

KATA PENGANTAR

Alhamdulillah, syukur yang tiada henti penulis panjatkan kehadirat Allah SWT serta tidak lupa sholawat serta salam semoga tetap tercurah kepada junjungan kita Nabi Muhammad SAW sehingga penelitian dalam tugas akhir ini bisa berjalan lancar dan selesai tepat pada waktunya.

Penulis ucapkan “Terima kasih” kepada Bapak Dr. Ir. Djoko Purwanto, M. Eng. dan Bapak Dr. Muhammad Rivai, ST., MT. selaku dosen pembimbing yang telah bersedia memberi pengarahan dan nasehat selama pengerjaan penelitian tugas akhir ini.

Penulis juga ucapkan “Terima Kasih” kepada para Dosen Penguji: Bapak Ir. Tasripan, MT., Bapak Astria Nur Irfansyah, ST., M. Eng., Bapak Ronny Mardiyanto, ST., MT., Ph. D., dan Bapak Fajar Budiman ST., M. Sc. yang telah menyediakan waktu untuk memberikan masukan dan pengarahan sehingga penulis dapat menyelesaikan buku tugas akhir ini dengan baik.

Tentu kepada kedua orangtua yang masih berjuang sekuat tenaga untuk putra beliau berdua ini, penulis ucapkan “Terimakasih dan Maaf”. Tidak lupa teman-teman di Laboratorium B.402-403 yang telah berjuang bersama selama penelitian tugas akhir ini, penulis ucapkan “Akhirnya. Terima Kasih Banyak”.

Penulis sadar bahwa Tugas Akhir ini masih belum sempurna dan masih banyak hal yang perlu diperbaiki. Saran, kritik dan masukan baik dari semua pihak sangat membantu penulis terutama untuk berbagai kemungkinan pengembangan lebih lanjut.

Terakhir, penulis berharap Tugas Akhir ini dapat memberikan manfaat bagi banyak pihak dan dapat membantu pengembangan tentang alat sentrifugasi kedepannya.

Surabaya, Juli 2018

Garudio Kusuma Aji

.....*Halaman ini sengaja dikosongkan*.....

DAFTAR ISI

PERNYATAAN KEASLIAN	i
ABSTRAK	i
<i>ABSTRACT</i>	iii
KATA PENGANTAR	v
DAFTAR ISI	vii
DAFTAR GAMBAR	ix
DAFTAR TABEL	xi
BAB I	1
PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Perumusan Masalah	3
1.3. Tujuan Tugas Akhir	3
1.4. Batasan Masalah	4
1.5. Sistematika	4
1.6. Metodologi	5
1.7. Relevansi	7
BAB II	8
TINJAUAN PUSTAKA	9
2.1. Alat Sentrifugasi	9
2.2. Logika Fuzzy	11
2.1.1. <i>Himpunan Fuzzy</i>	11
2.1.2. <i>Fungsi Keanggotaan (Membership Function)</i>	13
2.1.2.1. Representasi Linear	13
2.1.2.2. Representasi Kurva Segitiga	14
2.1.2.3. Representasi Kurva Trapesium	15
2.1.2.4. Representasi Kurva Bentuk Bahu	15
2.1.3. <i>Sistem Fuzzy Inferensi</i>	16
2.1.3.1. Metode Tsukamoto	17
2.1.3.2. Metode Mamdani	18
2.3. Mikrokontroler (Arduino Uno)	19
2.4. Motor DC	19
2.5. Motor Driver L298N	20
2.6. Sensor <i>Increment Encoder</i>	21
2.7. Tachometer DT-2234C+	22
2.8. <i>Real Time Clock</i>	23
2.9. <i>Buck Converter</i>	24
2.10. Komunikasi I2C	25

BAB III.....	27
PERANCANGAN ALAT SENTRIFUGASI	27
3.1. Perancangan Perangkat Keras.....	28
3.1.1. Sistem Antarmuka.....	28
3.1.2. Sistem Pewaktu pada Alat Sentrifugasi	28
3.1.3. Sistem Penggerak/Pemutar pada Alat Sentrifugasi	29
3.1.4. Sensor Kecepatan Sentrifugasi.....	30
3.2. Perancangan Perangkat Lunak	32
3.2.1. Sistem Antarmuka.....	32
3.2.2. Logika Fuzzy Set Point Kecepatan Sentrifugasi	35
3.2.3. Sensor Kecepatan Sentrifugasi.....	38
3.2.4. Logika Fuzzy Kestabilan Kecepatan Sentrifugasi	39
BAB IV.....	43
UJI COBA DAN ANALISIS.....	43
4.1. Pengujian Kecepatan Maksimal Alat Sentrifugasi.....	43
4.2. Pengujian Sensor Kecepatan Sentrifugasi.....	44
4.3. Pengujian <i>Duty Cycle</i> Terhadap Sistem Penggerak Alat Sentrifugasi	45
4.4. Pengujian Logika Fuzzy <i>Set Point</i> Kecepatan Senti- fugasi	46
4.5. Pengujian Logika Fuzzy Kestabilan Kecepatan Senti-fugasi.....	47
4.5.1. Pengujian Tanpa Beban	47
4.5.2. Pengujian Berbeban	49
4.6. Pengujian Keandalan Alat Sentrifugasi	51
BAB V	55
PENUTUP	55
5.1. Kesimpulan	55
5.2. Saran	55
DAFTAR PUSTAKA	57
LAMPIRAN A	59
LAMPIRAN B.....	61
LAMPIRAN C.....	65
LAMPIRAN D	71
LAMPIRAN E	81
BIODATA PENULIS	84

DAFTAR GAMBAR

Gambar 1.1 Alat sentrifugasi.	1
Gambar 1.2 Skema diagram blok pengendali kecepatan.	6
Gambar 2.1 Fenomena gaya sentripetal dan gaya sentrifugal.	9
Gambar 2.2 Fungsi keanggotaan linear naik.	13
Gambar 2.3 Representasi linear turun.	13
Gambar 2.4 Kurva representasi segitiga.	14
Gambar 2.5 Kurva representasi trapesium.	15
Gambar 2.6 Daerah ‘bahu’ pada variabel TEMPERATUR.	16
Gambar 2.6 Proses <i>fuzzy inference system</i>	16
Gambar 2.8 Inferensi metode Tsukamoto.	17
Gambar 2.9 <i>Mapping</i> pin dan terminal pada Arduino Uno.	19
Gambar 2.10 Struktur motor dcmp, tampak atas.	20
Gambar 2.11 <i>Module L298N</i>	20
Gambar 2.12 <i>Block diagram ic L298n dual full h-bridge</i>	21
Gambar 2.13 Sensor <i>incremental encoder</i> baris tunggal.	22
Gambar 2.14 Tachometer DT-2234C ⁺	22
Gambar 2.15 Modul RTC DS1307.	23
Gambar 2.16 Mini 360 DC-DC <i>Buck Converter</i>	24
Gambar 2.17 Skema komunikasi I ² C.	25
Gambar 3.1 Diagram blok sistem pengendali kecepatan alat sentrifugasi.	27
Gambar 3.2 Desain alat sentrifugasi.	27
Gambar 3.3 Skema sistem antarmuka alat sentrifugasi.	28
Gambar 3.4 Skema Sistem Pewaktu Alat Sentrifugasi.	28
Gambar 3.5 Skema Sistem Pemutar pada Alat Sentrifugasi.	29
Gambar 3.6 Skema sensor <i>increment encoder</i>	30
Gambar 3.7 Perancangan “piring” sentrifugasi.	31
Gambar 3.8 Skema tampilan menu awal.	31
Gambar 3.9 Skema tampilan menu ke-dua.	32
Gambar 3.10 Tampilan menu pada tahap proses sentrifugasi.	33
Gambar 3.11 Tampilan pada tahap terakhir.	33
Gambar 3.12a Diagram alir skema sistem antarmuka.	34
Gambar 3.12b Diagram alir skema sistem antarmuka.	35
Gambar 3.13 Diagram blok sistem logika <i>fuzzy set point</i> kecepatan.	35
Gambar 3.14 Diagram alir logika <i>fuzzy set point</i> kecepatan sentrifugasi.	36
Gambar 3.15 Grafik keanggotaan variabel konsentrasi.	37
Gambar 3.16 Grafik keanggotaan variabel waktu proses.	37

Gambar 3.17 Diagram alir sensor kecepatan sentrifugasi.	39
Gambar 3.18 Skema diagram blok logika fuzzy kestabilan kecepatan sentrifugasi.	39
Gambar 3.19 Diagram alir logika fuzzy kestabilan kecepatan sentrifugasi.	40
Gambar 3.20 Grafik keanggotaan variabel <i>error</i>	41
Gambar 3.21 Grafik keanggotaan variabel <i>deltaError</i>	42
Gambar 4.1 Pengujian kecepatan maksimal.	43
Gambar 4.2 Pengujian sensor <i>increment encoder</i>	45
Gambar 4.4 Respon logika fuzzy <i>set point</i> kecepatan sentrifugasi terhadap perubahan waktu.	47
Gambar 4.5 <i>Time respon</i> kecepatan <i>set point</i> 200 rpm.	48
Gambar 4.6 <i>Time respon</i> kecepatan <i>set point</i> 400 rpm.	48
Gambar 4.7 <i>Time respon</i> kecepatan <i>set point</i> 600 rpm.	48
Gambar 4.8 <i>Time respon</i> kecepatan <i>set point</i> 800 rpm.	49
Gambar 4.9 <i>Time respon</i> kecepatan <i>set point</i> 200 rpm.	49
Gambar 4.10 <i>Time respon</i> kecepatan <i>set point</i> 400 rpm.	50
Gambar 4.11 <i>Time respon</i> kecepatan <i>set point</i> 600 rpm.	50
Gambar 4.12 <i>Time respon</i> kecepatan <i>set point</i> 800 rpm.	50
Gambar 4.13 Kondisi santan referensi.	51
Gambar 4.14 Referensi dibandingkan sebelum sentrifugasi.	52
Gambar 4.15 Kondisi santan setelah disentrifugasi.	53

DAFTAR TABEL

Tabel 3.1 Skema fungsi <i>keypad</i> pada setiap tahap	32
Tabel 3.2 <i>Ruled Base Output, Set Point</i> Kecepatan Sentrifugasi	38
Tabel 3.3 <i>Ruled base output</i> koreksi <i>duty cycle</i> PWM.	42
Tabel 4.1 Kecepatan maksimal alat sentrifugasi tanpa beban.	43
Tabel 4.2 Kecepatan maksimal alat sentrifugasi berbeban.....	44
Tabel 4.3 Hasil uji sensor <i>increment encoder</i>	44
Tabel 4.4 Respon sistem penggerak alat sentrifugasi.	46
Tabel 4.5 Spesifikasi respon transien sistem.	51
Tabel 4.6 Spesifikasi santan referensi.	52
Tabel 4.7 Spesifikasi pemisahan santan hasil pengujian alat.	52
Tabel 4.8 Prosentase keberhasilan.	52

.....*Halaman ini sengaja dikosongkan*.....

BAB I

PENDAHULUAN

1.1. Latar Belakang

Alat sentrifugasi adalah suatu alat pemisah yang memanfaatkan perbedaan efek gaya sentrifugal pada setiap molekul senyawa penyusun suspensi dari gerak putar.

Gaya sentrifugal adalah gaya semu yang mendorong benda menjauhi titik pusat putar yang timbul pada suatu benda yang bergerak berputar pada kerangka non-inersia. Kerangka non-inersia pada alat sentrifugasi adalah botol tempat suspensi ditempatkan, botol menjaga agar suspensi tidak tumpah tetapi tidak mempertahankan posisi molekul-molekul senyawa penyusun suspensi. Efek gaya sentrifugal akan mendorong setiap molekul-molekul menjauhi titik pusat putar.^[1]

$$F_{\text{Sentrifugal}} = mv^2/R \quad (1.1)$$

Dimana F adalah gaya sentrifugal, m adalah massa objek, v adalah kecepatan linear dan R adalah jari-jari lintasan.

Berdasarkan rumus 1.1 besar nilai gaya sentrifugal pada suatu objek berbanding lurus dengan massa dan kecepatan linear objek, dan berbanding terbalik dengan besar jari-jari lintasannya.



Gambar 1.1 Alat sentrifugasi.

Pada alat sentrifugasi, gambar 1.1, besar kecepatan dan jari-jari lintasan setiap molekul senyawa pada suspensi sama besar, namun perbedaan massa setiap molekul mengakibatkan gaya sentrifugasi yang bekerja pada setiap molekul berbeda-beda. Semakin besar massa molekul senyawa semakin besar pula gaya sentrifugal yang bekerja pada molekul senyawa tersebut.

Perbedaan pengaruh besar gaya sentrifugal membuat senyawa penyusun suspensi terpisah, senyawa penyusun suspensi dengan massa molekul terbesar akan berada posisi terjauh dari titik pusat putar dan diikuti dengan senyawa penyusun suspensi dengan massa molekul yang lebih kecil, sampai senyawa penyusun suspensi dengan massa molekul terkecil. Senyawa penyusun suspensi dengan massa terkecil akan berada pada posisi terdekat titik pusat putar.

Dalam bidang kesehatan, alat sentrifugasi sering digunakan untuk pemisahan darah. Darah yang tersusun dari plasma darah, sel darah merah, sel darah putih, dan keping darah perlu dipisahkan untuk menyesuaikan kebutuhan pasien. Dalam bidang industri makanan, alat sentrifugasi digunakan pada pengolahan susu skimmed, susu rendah lemak. Susu yang mengandung lemak disentrifugasikan, efek gaya sentrifugasi akan membuat lemak pada susu “terangkat” ke permukaan, mendekati titik pusat putar. Hal ini dikarenakan massa molekul lemak lebih kecil daripada massa molekul air pada susu. Dalam bidang industri UMKM, alat sentrifugasi digunakan pada proses pembuatan *Virgin Coconut Oil (VCO)*.

VCO adalah minyak kelapa yang dihasilkan dengan sedikit proses pemanasan. Terdapat dua metode pemisahan dalam pembuatan VCO, metode pendiaman dan metode sentrifugasi. Metode pendiaman, santan didiamkan, memanfaatkan efek gaya gravitasi. Metode Sentrifugasi, santan dimasukkan dalam wadah alat sentrifugasi, memanfaatkan efek gaya sentrifugal.

Alat pemisah atau sentrifugasi yang banyak digunakan oleh para pengusaha VCO belum bisa dikatakan efisien dari sisi kecepatan sentrifugasi. Spesifikasi kecepatan putar sentrifugasi yang didasarkan pada spesifikasi kecepatan putar motor, bukanlah cara yang tepat. Karena besar massa/berat beban yang diputar motor saat proses sentrifugasi akan mempengaruhi kecepatan putar motor.

Selain itu alat sentrifugasi konvensional berputar dengan kecepatan penuh dari awal proses sampai dengan akhir proses, tanpa ada penyesuaian kecepatan dengan kondisi suspensi santan yang dipisahkan. Kondisi yang dimaksudkan diantaranya:

- kondisi suspensi santan masih tercampur,

- kondisi suspensi santan sudah mulai terpisah,
- kondisi suspensi santan sudah terpisah,
- kondisi suspensi santan sudah banyak yang terpisah,
- kondisi suspensi santan sudah mencapai titik jenuh pemisahan,

Contoh kasus, pada saat keadaan motor bekerja dengan kecepatan penuh, namun kondisi santan sudah mencapai titik jenuh pemisahan, hal ini tidak efisien, dikarenakan efek gaya sentrifugal yang ditimbulkan motor sudah tidak berpengaruh pada suspensi santan.

Penambahan kontroler pengendali kecepatan yang mampu menyesuaikan dengan kondisi suspensi diharapkan mampu meningkatkan efisiensi alat sentrifugasi.

Kontroler logika fuzzy adalah salah satu metode kontroler yang sedang berkembang saat ini. Logika fuzzy merupakan metode kontroler yang memungkinkan proses secara penalaran bahasa, sehingga dalam perancangan sistem kendali/kontrol tidak memerlukan persamaan matematik dari objek yang akan dikendalikan.^[2]

Kontroler logikafuzzy akan diimplementasikan dalam pengendali kecepatan alat sentrifugasi untuk meningkatkan efisiensi pemisahan *Virgin Coconut Oil* dari suspensi santan

Pada tugas akhir ini akan dirancang alat sentrifugasi yang dilengkapi pengendali/pengaturan kecepatan yang bertujuan untuk meningkatkan kehandalan alat sentrifugasi dengan mengendalikan kecepatan sentrifugasi sesuai dengan kondisi dari suspensi santan dalam waktu tertentu.

1.2. Perumusan Masalah

Permasalahan-permasalahan yang dibahas dalam Tugas Akhir ini dalam merancang alat sentrifugasi diantaranya:

1. Bagaimana karakteristik respon suspensi santan terhadap perubahan variabel kecepatan, waktu, dan, konsentrasi.
2. Bagaimana merancang Logika Fuzzy untuk menentukan *set point* kecepatan sentrifugasi yang sesuai dengan data karakteristik.
3. Bagaimana merancang kontroler kestabilan kecepatan sentrifugasi dengan metode Logika Fuzzy.

1.3. Tujuan Tugas Akhir

Tujuan-tujuan dari perancangan alat sentrifugasi pada tugas akhir ini, diantaranya:

1. Mengatahui karakteristik respon suspensi santan terhadap perubahan variabel kecepatan, waktu, dan, konsentrasi.
2. Mampu merancang Logika Fuzzy untuk menentukan *set point* kecepatan sentrifugasi yang sesuai dengan data karakterisasi.
3. Mampu merancang kontroler kestabilan kecepatan sentrifugasi dengan Logika Fuzzy.

1.4. Batasan Masalah

Pada alat sentrifugasi tugas akhir ini terdapat batasan-batasan berkaitan dengan kapasitas daya tampung, kecepatan sentrifugasi, waktu sentrifugasi, dan beberapa hal lainnya. Berikut batasan-batasan alat sentrifugasi pada tugas akhir ini:

1. Uji coba dilakukan di Laboratorium Elektronika Industri, B.402, Departemen Teknik Elektro Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember.
2. Uji coba dilakukan pada suhu ruangan Laboratorium Elektronika Industri, B.402.
3. Pada alat sentrifugasi terdapat tiga (3) botol untuk wadah suspensi santan, dengan kapasitas setiap botol 50 gram.
4. Kecepatan putar maksimal alat sentrifugasi adalah 800 rpm.
5. Waktu maksimal setiap proses sentrifugasi adalah 30 menit.

Kondisi diatas adalah batasan-batasan mengenai alat sentrifugasi yang dirancang pada tugas akhir ini.

1.5. Sistematika

Berikut sistematika penulisan Tugas Akhir Alat Sentrifugasi:

- **BAB I : Pendahuluan**
Pada bab ini membahas latar belakang, perumusan masalah, tujuan penulisan, sistematika penulisan, dan Relevansi alat sentrifugasi yang dirancang pada Tugas Akhir ini.
- **BAB II : Tinjauan Pustaka**
Pada bab ini membahas tentang teori-teori penunjang dalam merancang alat sentrifugasi, diantaranya:
 - Alat Sentrifugasi
 - Logika Fuzzy
 - Mikrokontroler (Arduino)
 - Komunikasi I2C
 - Motor DC
 - Motor Driver L298N

- Sensor *Increment Encoder*
- Tachometer DT-2234C⁺
- BAB III : Metodologi Perancangan Alat
 Pada bab ini membahas tentang bagaimana perancangan sistem pengatur kecepatan alat sentrifugasi pada Tugas Akhir ini, diantaranya:
 - Perancangan Perangkat Keras
 - Perancangan Perangkat Lunak
- BAB IV : Uji Coba dan Analisis Data
 Pada bab ini membahas tentang uji coba yang dilakukan dari hasil perancangan pengendali kecepatan alat sentrifugasi berbasis kontroler Logika Fuzzy. Uji coba yang dilakukan diantaranya:
 - Pengujian Sensor Kecepatan Putar
 - Pengujian Kecepatan Maksimal Alat Sentrifugasi
 - Pengujian Sistem Logika Fuzzy *Set Point* Kecepatan Sentrifugasi
 - Pengujian Sistem Logika Fuzzy Kestabilan Kecepatan Sentrifugasi
 - Karakterisasi Efek Sentrifugasi pada Santan terhadap Perubahan Variabel Waktu, Konsentrasi, dan Kecepatan
 - Pengujian Pengendali Kecepatan Alat Sentrifugasi

Seluruh hasil pengujian tersebut dianalisis untuk mengetahui keandalan hasil perancangan Pengendali Kecepatan Alat Sentrifugasi.

- BAB V : Penutup
 Pada bab ini membahas tentang kesimpulan-kesimpulan yang berkaitan tentang proses perancangan pengendali kecepatan dengan metode Logika Fuzzy dan perbandingan kehandalan, efektifitas dan efisiensi, alat sentrifugasi hasil rancangan dengan alat sentrifugasi konvensional, tanpa kontroler. Serta saran-saran yang dibutuhkan untuk mengembangkan alat sentrifugasi.

1.6. Metodologi

Berikut ini diagram alir tahapan-tahapan penyusunan Tugas Akhir ini dan penjelasan setiap tahapan:

- **Studi literatur**
 Pada tahap ini dilakukan pengumpulan dasar teori yang menunjang dalam penulisan Tugas Akhir. Dasar teori ini dapat diambil

dari buku-buku, jurnal, *proceeding*, dan artikel-artikel di internet dan forum-forum diskusi internet.

- **Studi Lapangan**

Pada tahap ini akan dilakukan survei di lapangan untuk mengumpulkan informasi yang dapat mendukung, menyeleksi, dan menambahkan bahan dari studi literatur.

- **Perancangan Pengatur Kecepatan Alat Sentrifugasi**

Tahap ini terbagi menjadi dua bagian:

- a. **Perancangan Komponen Keras (Antarmuka, Mikrokontroler, Aktuator, Sensor)**

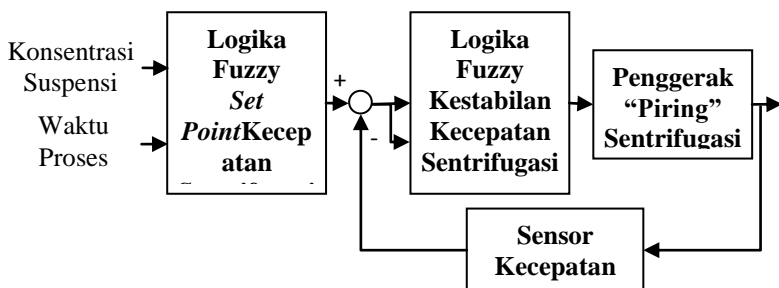
Pemilihan *hardware* untuk membangun/merakit alat sentrifugasi sesuai dengan informasi yang telah didapatkan pada tahap *Studi Literature* dan lapangan.

Keypad dan *Liquid Crystal Display* (LCD) sebagai perangkat antarmuka, Arduino Uno R3 sebagai mikrokontroler, motor DC 775 sebagai aktuator, dan sensor *encoder increment* sebagai sensor kecepatan.

- b. **Perancangan Komponen Lunak (Algoritma Antarmuka dan Logika Fuzzy)**

Pada tahap ini merancang alat pengatur kecepatan rotasi pada alat sentrifugasi dengan kontrol *fuzzy*.

Variabel konsentrasi dan pewaktu sebagai input kontroler logika fuzzy kecepatan *Set Point* sentrifugasi. Perbedaan, *error*, kecepatan *Set Point* dengan kecepatan pembacaan sensor kecepatan dan perbedaan dari *error* saat ini dengan *error*, *deltaError*, sebelumnya dijadikan sebagai input kontroler logika fuzzy kestabilan kecepatan sentrifugasi.



Gambar 1.2 Skema diagram blok pengendali kecepatan.

- **Pengujian Alat Sentrifugasi**

Pengujian dilakukan secara bertahap. Pengujian masing-masing bagian subsistem alat sentrifugasi dilakukan agar pemecahan masalah lebih mudah dan sederhana. Berikut urutan pengujian pada tugas akhir ini:

- Pengujian Sensor Kecepatan Putar
- Pengujian Kecepatan Maksimal Alat Sentrifugasi
- Pengujian Sistem Logika Fuzzy *Set Point* Kecepatan Sentrifugasi
- Pengujian Sistem Logika Fuzzy Kestabilan Kecepatan Sentrifugasi
- Pengujian Keandalan Alat sentrifugasi

Pengujian dilakukan untuk mendapatkan keandalan dari alat yang telah direalisasikan.

- **Penulisan Buku Tugas Akhir**

Tahap penulisan laporan Tugas Akhir dilakukan pada saat tahap pengujian sistem dimulai serta setelahnya.

1.7. Relevansi

Hasil dari Tugas Akhir ini diharapkan dapat membantu meningkatkan keandalan alat sentrifugasi produksi para pengusaha UMKM di bidang industri terkait.

.....*Halaman ini sengaja dikosongkan*.....

BAB II

TINJAUAN PUSTAKA

2.1. Alat Sentrifugasi

Alat sentrifugasi merupakan alat pemisah yang digunakan untuk memisahkan campuran padat/cair atau cair/cair yang saling tidak larut yang memanfaatkan gaya sentrifugal pada kecepatan putar tinggi. Prinsip sentrifugasi didasarkan atas fenomena bahwa partikel yang tersuspensi di dalam suatu wadah akan mengendap ke dasar wadah karena pengaruh gravitasi. Laju pengendapan tersebut dapat ditingkatkan dengan cara meningkatkan pengaruh gravitasional terhadap partikel, hal ini dapat dilakukan dengan menempatkan wadah berisi suspensi partikel ke dalam rotor suatu mesin sentrifugasi kemudian diputar dengan kecepatan tinggi.

Jika gaya F diterapkan pada suatu partikel dengan massa m , maka partikel akan dipercepat dengan arah linear sehingga:

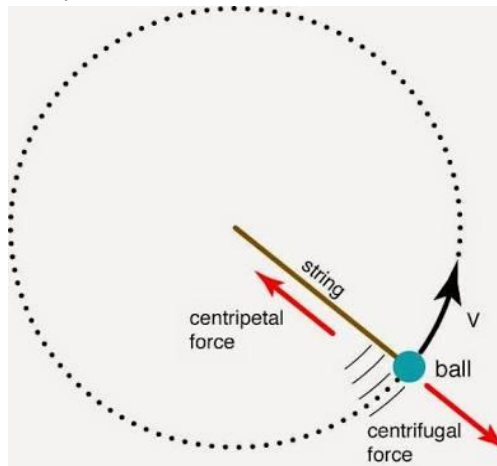
$$F = m \cdot a \quad (2.1)$$

a adalah laju percepatan linear, akan tetapi yang terjadi pada sentrifugasi adalah percepatan *angular*, sehingga:

$$a = v^2 / x \quad (2.2)$$

Dimana v adalah kecepatan linear dan x adalah jari-jari lintasan.

$$F = m \cdot v^2 / x \quad (2.3)$$



Gambar 2.1 Fenomena gaya sentripetal dan gaya sentrifugal.

Pada umumnya nilai yang diberikan untuk gaya yang berlaku pada partikel yang sentrifugasi berupa nilai relatif, yaitu dibandingkan dengan gaya tarik gravitasi bumi yang juga berlaku pada partikel tersebut. Gaya tersebut disebut gaya sentrifugasi relatif (*relative centrifugal force*, RCF). Gaya gravitasi yang berlaku pada partikel dengan massa m adalah:

$$F = m \cdot g \quad (2.4)$$

g adalah percepatan yang besarnya 980 cm/detik². Dengan demikian:

$$RCF = F_{\text{sentrifugasi}} / F_{\text{gravitasi}} \quad (2.5)$$

Dengan perhitungan secara matematis yang tidak dijelaskan di sini, maka nilai RCF adalah:

$$RCF = (1,119 \times 10^{-5} (2\pi \text{rad/minute})^{-2} (\text{cm})^{-1} (v^2)(r)) \quad (2.6)$$

Dimana v adalah kecepatan putaran per menit (rpm) dan r adalah jari-jari alat sentrifugasi

Kecepatan proses pengendapan (sedimentasi) suatu partikel atau molekul yang disentrifugasi dipengaruhi dua faktor, yaitu:

1. Berat molekul (BM). Semakin tinggi BM molekul maka kecepatan pengendapan molekul juga semakin tinggi.
2. Bentuk partikel. Gerakan suatu partikel melalui cairan akan dipengaruhi oleh gaya gesekan. Partikel yang mempunyai bentuk lebih kompak akan bergerak lebih cepat di dalam cairan dibandingkan dengan partikel lain yang bentuknya kurang kompak meskipun memiliki BM yang sama. Dalam pembahasan mengenai proses sedimentasi dikenal suatu konstanta yang disebut **koefisien sedimentasi** (s) yang nilainya:

$$s = \text{Kecepatan Molekular} / \text{Gaya Sentrifugal} \quad (2.7)$$

Nilai suatu partikel menunjukkan karakteristiknya di dalam suatu medium (larutan) pada suhu tertentu. Nilai umumnya sama pada larutan yang berbeda sehingga s juga dinyatakan dalam satuan Svedberg (S) yang besarnya $1 \text{ S} = 10^{-13}$ detik yang berlaku dalam kondisi standar yaitu dalam air pada suhu 20°C. Koefisien sedimentasi makromolekul dalam sel berkisar antara 1 S sampai 200 S. Svedberg adalah ahli biokimia pemenang hadiah Nobel yang mengembangkan ultrasentrifugasi.

2.2. Logika Fuzzy

Logika fuzzy merupakan salah satu cabang dari bidang *soft computing*. Logika fuzzy pertama kali diperkenalkan pada tahun 1965 oleh Lotfi A. Zadeh, seorang profesor ilmu komputer di Universitas of California di Berkeley. Logika fuzzy merupakan suatu teori himpunan logika yang dikembangkan untuk mengatasi konsep nilai yang terdapat diantara kebenaran (*true*) dan kesalahan (*false*). Zadeh menyatakan bahwa setiap persoalan dapat diselesaikan tanpa menggunakan logika *fuzzy*, tetapi dengan menggunakan logika fuzzy akan mempercepat dan mempermudah penyelesaian suatu persoalan. Logika fuzzy berbeda dengan logika digital biasa atau *Boolean*. Logika digital biasa hanya mengenal dua keadaan yang tegas (*crisp*), yaitu: 'ya' atau 'tidak', '0' atau '1', dan 'On' atau 'Off'. Berbeda dengan logika digital biasa, logika *fuzzy* meniru cara berfikir manusia dengan menggunakan konsep sifat kesamaan suatu nilai. Dengan menggunakan logika fuzzy, nilai tidak lagi hanya bernilai '0' dan '1', tetapi seluruh kemungkinan diantara 0 sampai 1. Contoh penerapan fuzzy pada aplikasi yaitu pengontrol suhu ruangan, prediksi cuaca, pengelolaan keuangan, dan masih banyak lainnya.

Kesederhanaan konsep membuat konsep logika fuzzy mudah dimengerti. Fuzzy tidak terpaku pada satu keputusan, fleksibel, sehingga dapat memberi nilai toleransi pada ketidakpastian dan fuzzy disusun berdasarkan bahasa manusia sehingga tidak sulit dalam memahaminya. Ada beberapa alasan mengapa memilih menggunakan logika fuzzy yaitu:^[2]

- Konsep logika fuzzy mudah dimengerti. Konsep matematis dari logika yang sangat sederhana.
- Sifat logika fuzzy yang fleksibel.
- Logika fuzzy mampu menggambarkan fungsi-fungsi linear yang bersifat kompleks.

2.1.1. Himpunan Fuzzy

Pada himpunan *crisp*, nilai keanggotaan hanya ada dua, 0, kemungkinan, yaitu 0 atau 1, pada himpunan fuzzy nilai keanggotaan terletak pada rentang 0 sampai 1. Apabila x memiliki nilai keanggotaan fuzzy $\mu_A[x] = 0$ berarti x tidak menjadi anggota himpunan A , demikian pula apabila x memiliki nilai keanggotaan fuzzy $\mu_A[x] = 1$ berarti x menjadi anggota penuh pada himpunan A .

Terkadang kemiripan antara keanggotaan fuzzy dengan probabilitas menimbulkan kerancuan. Keduanya memiliki nilai pada interval $[0,1]$, namun interpretasi nilainya sangat berbeda antara kedua kasus tersebut. Kenggotaan fuzzy memberikan suatu ukuran terhadap pendapat atau keputusan, sedangkan probabilitas mengindikasikan proporsi terhadap keseringan suatu hasil bernilai benar dalam jangka panjang. Misalnya, jika nilai kenggotaan suatu himpunan fuzzy MUDA adalah 0,9; maka tidak perlu dipemalahkan berapa seringnya nilai itu diulang secara individual untuk mengharapkan suatu hasil yang hampir pasti muda. Di lain pihak, nilai probabilitas 0,9 muda berarti 10% dari himpunan tersebut diharapkan tidak muda.

Himpunan fuzzy memiliki 2 atribut, yaitu:

- a. Linguistik, yaitu penamaan suatu grup yang mewakili suatu keadaan atau kondisi tertentu dengan menggunakan bahasa alami, seperti: MUDA, PAROBAYA, TUA.
- b. Numeris, yaitu suatu nilai (angka) menunjukkan ukuran dari suatu variabel seperti: 40, 25, 50, dan sebagainya.

Ada beberapa hal yang perlu diketahui dalam memahami sistem fuzzy, yaitu:

1. Variabel fuzzy

Variabel fuzzy merupakan variabel yang hendak dibahas dalam suatu sistem fuzzy. Contoh: umur, temperatur, permintaan, dan sebagainya.

2. Himpunan Fuzzy

Himpunan fuzzy merupakan suatu grup yang mewakili suatu kondisi atau keadaan tertentu dalam suatu variabel fuzzy.

3. Semesta Pembicaraan

Semesta pembicaraan adalah keseluruhan nilai yang diperbolehkan untuk dioperasikan dalam suatu variabel fuzzy. Semesta pembicaraan merupakan himpunan bilangan real yang senantiasa naik (bertambah) secara monoton dari kiri ke kanan. Nilai semesta pembicaraan dapat berupa bilangan positif maupun negatif. Adakalanya nilai semesta pembicaraan ini tidak dibatasi atasnya.

4. Domain

Domain himpunan fuzzy adalah keseluruhan nilai yang diijinkan dalam semesta pembicaraan dan boleh dioperasikan dalam suatu himpunan fuzzy. Seperti halnya semesta pembicaraan, domain merupakan himpunan bilangan real yang senantiasa naik (bertambah) secara monoton dari kiri ke kanan. Nilai domain dapat berupa bilangan positif maupun negatif.

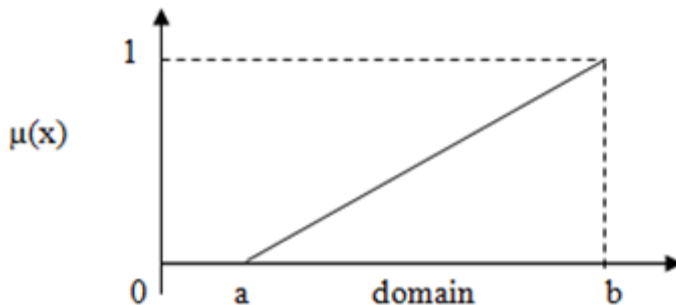
2.1.2. Fungsi Keanggotaan (*Membership Function*)

Fungsi keanggotaan (*membership function*) adalah suatu kurva yang menunjukkan pemetaan titik-titik input data ke dalam nilai keanggotaannya (sering juga disebut derajat keanggotaan) yang memiliki interval antara 0 sampai 1. Salah satu cara yang dapat digunakan untuk mendapatkan nilai keanggotaan adalah dengan melalui pendekatan fungsi. Ada beberapa fungsi yang bisa digunakan.

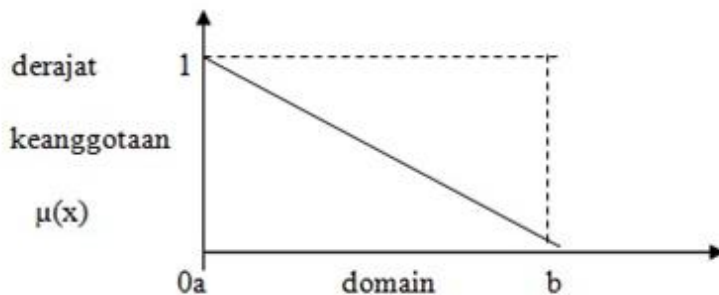
2.1.2.1. Representasi Linear

Pada representasi linear, pemetaan input ke derajat keanggotaannya digambarkan sebagai suatu garis lurus. Bentuk ini paling sederhana dan menjadi pilihan yang baik untuk mendekati suatu konsep yang kurang jelas.

Ada 2 keadaan himpunan fuzzy yang linear. Pertama, kenaikan himpunan dimulai pada nilai domain yang memiliki derajat keanggotaan nol [0] bergerak ke kanan menuju ke nilai domain yang memiliki derajat ke anggotaan lebih tinggi. (Gambar 2.2)



Gambar 2.2 Fungsi keanggotaan linear naik.



Gambar 2.3 Representasi linear turun.

Fungsi keanggotaan:

$$\mu[x] = \begin{cases} 0; & x \leq 0 \\ (x - a)/(x - b); & a \leq x \leq b \\ 1; & x \geq b \end{cases} \quad (2.8)$$

Kedua, merupakan kebalikan yang pertama. Garis lurus di mulai dari nilai domain dengan derajat keanggotaan tertinggi pada sisi kiri, kemudian bergerak menurun ke nilai domain yang memiliki derajat keanggotaan lebih rendah (Gambar 2.3).

Fungsi keanggotaan:

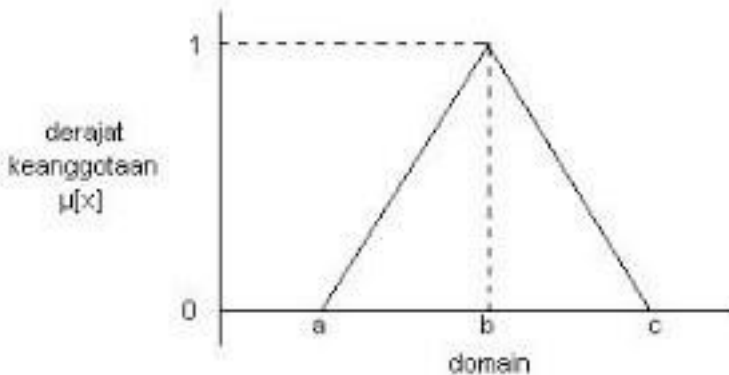
$$\mu[x] = \begin{cases} (b - x)/(b - a); & a \leq x \leq b \\ 0; & x \geq b \end{cases} \quad (2.9)$$

2.1.2.2. Representasi Kurva Segitiga

Kurva segitiga adalah kombinasi dari dua garis (linear) seperti terlihat pada Gambar 2.4:

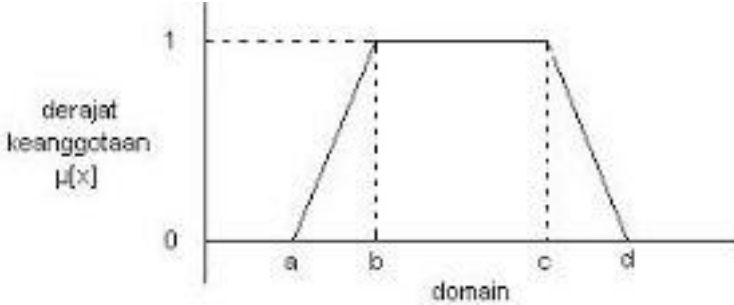
Fungsi keanggotaan:

$$\mu[x] = \begin{cases} 0; & x \leq a \\ (x - a)/(b - a); & a \leq x \leq b \\ (b - x)/(c - b); & b \leq x \leq c \\ 0; & x \geq c \end{cases} \quad (2.10)$$



Gambar 2.4 Kurva representasi segitiga.

2.1.2.3. Representasi Kurva Trapesium



Gambar 2.5 Kurva representasi trapesium.

Kurva trapesium pada dasarnya seperti bentuk segitiga, dimana terdapat beberapa titik yang memiliki nilai keanggotaan 1. (Gambar 2.5)

Fungsi keanggotaan:

$$\mu[x] = \begin{cases} 0; & x \leq a \\ (x - a)/(b - a); & a \leq x \leq b \\ 1; & b \leq x \leq c \\ (c - x)/(d - c); & c \leq x \leq d \\ 0; & x \geq d \end{cases} \quad (2.11)$$

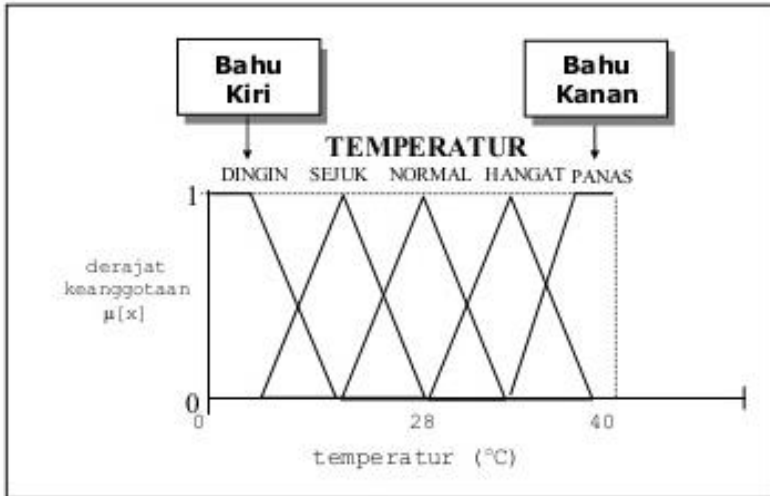
2.1.2.4. Representasi Kurva Bentuk Bahu

Daerah yang terletak ditengah-tengah suatu variabel yang direpresentasikan dalam bentuk segitiga, pada sisi kanan dan kirinya akan naik dan turun. Tetapi terkadang salah satu sisi dari variabel tersebut tidak mengalami perubahan. Sebagai contoh, apabila telah mencapai kondisi PANAS, kenaikan temperatur akan tetap berada pada kondisi PANAS. Himpunan fuzzy ‘bahu’, bukan segitiga, digunakan untuk mengakhiri variabel suatu daerah fuzzy. Bahu kiri bergerak dari 1 ke 0, sementara bahu kanan bergerak dari 0 ke 1. Gambar 2.6 menunjukkan variabel TEMPERATUR dengan daerah bahunya.

Fungsi keanggotaan:

$$\mu_{dingin} [x] = \begin{cases} 1; & a \leq x \leq b \\ (c - x)/(c - b); & b \leq x \leq c \\ 0; & x \geq c \end{cases} \quad (2.12)$$

$$\mu_{sejuk} [x] = \begin{cases} 0; & x \leq b \\ (x - b)/(c - b); & b \leq x \leq c \\ (d - x)/(d - c); & c \leq x \leq d \\ 0; & x \geq d \end{cases} \quad (2.13)$$



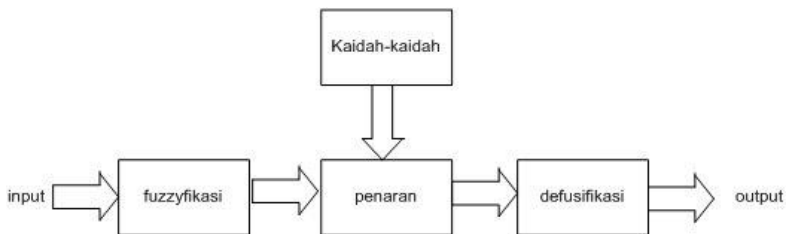
Gambar 2.6 Daerah ‘bahu’ pada variabel TEMPERATUR.

$$\mu_{normal} [x] = \begin{cases} 0; & x \leq c \\ (x - c)/(d - c); & c \leq x \leq d \\ (e - x)/(e - d); & d \leq x \leq e \\ 0; & x \geq e \end{cases} \quad (2.14)$$

$$\mu_{hangat} [x] = \begin{cases} 0; & x \leq d \\ (x - d)/(e - d); & d \leq x \leq e \\ (f - x)/(f - e); & e \leq x \leq f \\ 0; & x \geq f \end{cases} \quad (2.15)$$

$$\mu_{panas} [x] = \begin{cases} 0; & x \leq e \\ (x - e)/(f - e); & e \leq x \leq f \\ 1; & x \geq f \end{cases} \quad (2.16)$$

2.1.3. Sistem Fuzzy Inferensi



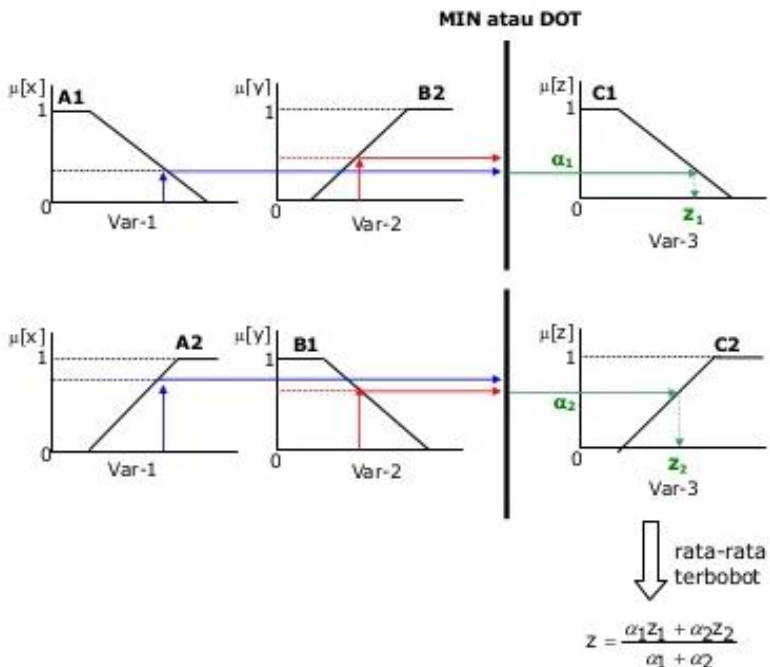
Gambar 2.6 Proses fuzzy inference system.

Sistem inferensi fuzzy (*Fuzzy Inference System/FIS*) adalah sistem yang dapat melakukan penalaran dengan prinsip seperti manusia melakukan penalaran dengan nalurinya.

Proses dalam FIS pada gambar 2.7 input yang diberikan kepada FIS adalah berupa bilangan tertentu dan output yang dihasilkan jugaberupa bilangan tertentu.Terdepat beberapa jenis FIS yang dikenal yaitu mamdani, Sugeno dan Tsukamoto.

2.1.3.1. Metode Tsukamoto

Pada metode Tsukamoto, setiap konsekuen pada aturan yang terbentuk IF-THEN harus direpresentasikan dengan suatu himpunan fuzzy dengan fungsi keanggotaan yang monoton. Sebagai hasilnya, *output* hasil interfensi dari setiap aturan diberikan secara tegas (*crisp*) berdasarkan a-predikat (*fire strength*). Hasil akhirnya diperoleh dengan menggunakan rata-rata terbobot.



Gambar 2.8 Inferensi metode Tsukamoto.

2.1.3.2. Metode Mamdani

Dalam metode Mamdani, proses implikasi fuzzy menggunakan operator min atau *conjunction* yang menandakan nilai minimum, sedangkan aturan agresi menggunakan operator max yang merupakan nilai maksimumnya.

Metode Mamdani memiliki empat tahapan di dalam melakukan proses terhadap variabel input. Keempat tahapan tersebut ialah:

1. Fuzzification

Tahapan dalam metode mamdani yaitu mengubah *crisp input* menjadi angka fuzzy, dan meletakkan pada *fuzzy set* yang benar.

2. Rule Evaluation

Setelah dilakukan fuzzifikasi, maka aturan yang berlaku akan dievaluasi bersamaan dengan *fuzzy set* yang sudah didapat.

Jika kondisinya menggunakan operasi “AND” maka rumusnya:

$$\mu_{A \cap B}(x) = \min(\mu_A[x], \mu_B[x]) \quad (2.17)$$

Jika kondisi menggunakan operasi “OR” maka rumusnya:

$$\mu_{A \cup B}(x) = \max(\mu_A[x], \mu_B[x]) \quad (2.18)$$

Dimana u adalah *fuzzy set*, $\mu_A[x]$ adalah kumpulan *fuzzy set* A dari x dan $\mu_B[x]$ adalah kumpulan *fuzzy set* B dari x.

3. Rule Aggregation

Tahap ini akan dilakukan agregasi pada *output set* yang mengambil nilai maksimumnya dengan menggunakan fungsi max.

$$\mu_{sf}[x_i] \leftarrow \max(\mu_{kf}[x_f], \mu_{kf}[x_i]) \quad (2.19)$$

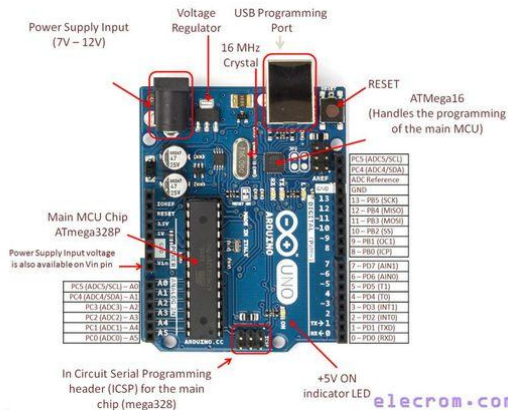
Dangan μ_{sf} merupakan fuzzy set.

4. Defuzzification

Setelah dilakukan agregasi, maka selanjutnya adalah mengubah *fuzzy output* menjadi *crisp output*. Metode *defuzzification* yang digunakan adalah metode centroid atau sering disebut dengan *center of gravity* (COG).

$$COG = \frac{\sum_{x=a}^b \mu_A(x)x}{\sum_{x=a}^b \mu_A(x)} \quad (2.20)$$

2.3. Mikrokontroler (Arduino Uno)



Gambar 2.9 Mapping pin dan terminal pada Arduino Uno.

Arduino adalah sebuah *board* mikrokontroler yang bersifat *open source*, dimana desain skematik dan PCB bersifat *open source*, sehingga *user* dapat menggunakannya maupun melakukan modifikasi. *Board* Arduino menggunakan *Chip/IC* mikrokontroler Atmel AVR.^[3]

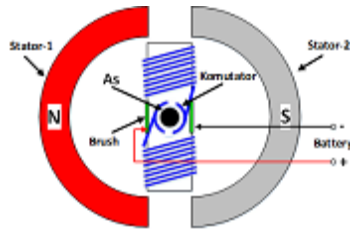
Board Arduino Uno menggunakan mikrokontroler Atmega328. Secara umum posisi/letak pin-pin terminal I/O pada berbagai *board* Arduino hampir sama dengan posisi pin-pin terminal I/O dari Arduino Uno yang mempunyai 14 pin Digital yang dapat diatur sebagai *input* atau *output* dan 6 pin *input* Analog.

2.4. Motor DC

Motor listrik DC atau DC Motor adalah suatu perangkat yang mengubah energi listrik arus searah menjadi energi kinetik atau gerakan (*motion*). Motor DC juga dapat disebut dengan motor arus searah. DC motor memiliki dua terminal dan memerlukan tegangan arus searah atau DC (*Direct Current*) untuk dapat mengerakkannya.

Arah putaran yang dihasilkan motor dc dapat diubah searah dengan putaran jarum jam atau berlawanan arah dengan putaran jarum jam. Perubahan ini dapat dilakukan hanya dengan mengubah polaritas listrik yang diberikan pada motor DC.^[4]

Motor DCMP adalah motor DC yang menggunakan magnet permanen pada motor untuk mengubah energi listrik menjadi energi kinetik.



Gambar 2.10 Struktur motor dcmp, tampak atas.

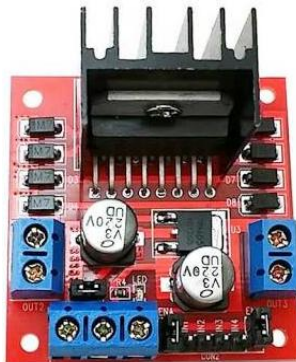
Dari gambar 2.10 dapat diketahui bahwa pada motor DCMP memiliki bagian-bagian sebagai berikut:

Bagian Rotor, rotor yang berupa kumparan listrik yang digulung dalam suatu inti besi yang dirangkai dengan As motor. Kemudian pada bagian ujung kumparan terdapat sepasang sikat (brush) yang berfungsi mengalirkan tegangan listrik DC (+ dan -) dari sumber tegangan menuju kumparan listrik untuk menghasilkan medan elektromagnet pada kumparan listrik.

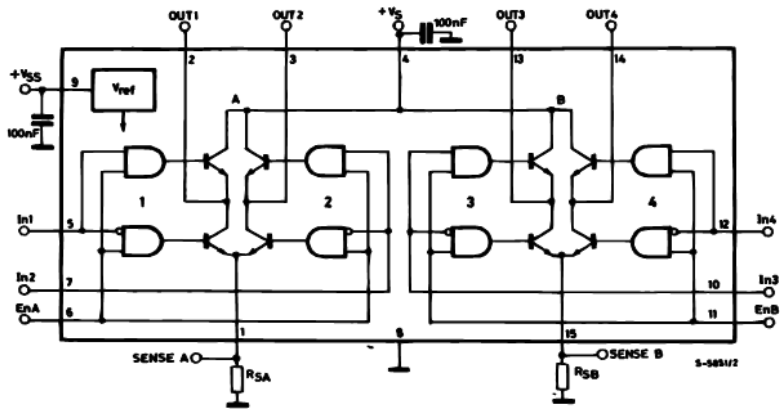
Bagian Stator, berbeda dengan tipe motor DC sebelumnya dua stator pada motor DCMP berupa magnet permanen dengan kutub yang berbeda, yaitu satu magnet ber kutub selatan (S) dan satu magnet ber kutub utara (N).

2.5. Motor DriverL298N

Module L298N adalah modul dual H-Bridge L298N adalah sebuah rangkaian terintegrasi tipe monolitik. L298N merupakan *high voltage, high current dual full-bridge driver* yang dirancang untuk tingkat logika IC TTL standar dan mengendalikan beban-beban yang bersifat induktif, seperti: relay, solenoid, motor DC dan motor Stepper.^[7]



Gambar 2.11Module L298N.



Gambar 2.12Block diagram ic l298n dual full h-bridge.

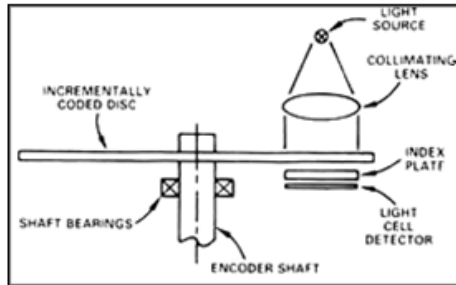
Terdapat dua kaki *enable input*, Enable A dan Enable B, yang memungkinkan untuk mengaktifkan atau menonaktifkan perangkat secara independen dari sinyal input. Kaki-kaki emitter transistor bawah (lihat gambar 2.12) dari setiap *bridged* disatukan sehingga terminal eksternal dapat digunakan untuk koneksi dengan resistor sensing eksternal.

Berikut spesifikasi dari modul *motor driver* L298N:

- *Double H Bridge Drive Chip* : L298N
- *Logical Voltage* : 5V
- *Drive Voltage* : 5V-35V
- *Logical Current* : 0-36mA
- *Drive Current* : 2A (MAX single bridge)
- Daya Maksimal : 25W
- Ukuran : 43 × 43 × 26mm
- Berat : 26 g

2.6. Sensor Increment Encoder

Incremental encoder bisa digunakan *feedback* posisi dan kecepatan motor. *Incremental encoder* menghasilkan *feedback* kecepatan dan jarak objek dengan sangat baik, karena hanya sedikit sensor yang menyusunnya, sistem *incremental encoder* simpel dan murah. *Incremental encoder* hanya terbatas menghasilkan perubahan HIGH atau LOW, sehingga encoder membutuhkan perangkat tambahan untuk menghitung kecepatan atau posisi objek.



Gambar 2.13 Sensor *incremental encoder* baris tunggal.

Incremental encoder menghasilkan jumlah pulsa tertentu dalam satu putaran objek. Keluaran *encoder* bisa baris pulsa tunggal atau ganda. *Incremental encoder* baris pulsa tunggal mampu digunakan untuk menghitung kecepatan objek, namun tidak mampu menentukan arah putaran objek. *Incremental encoder* baris pulsa ganda mampu digunakan untuk menghitung kecepatan dan arah putaran objek.

2.7. Tachometer DT-2234C⁺

Tachometer DT-2234C⁺ adalah tachometer digital yang menggunakan sinar LED dan sinar laser sebagai pendeteksi.



Gambar 2.14 Tachometer DT-2234C⁺.

Berikut spesifikasi dari Tachometer DT-2234C⁺:

- Tampilan : 5digit, 18 mm, LCD.
- Jangkauan Uji : 2,5 sampai 99,999 rpm.
- Resolusi : 0,1 rpm (2,5-999,9 rpm),
1 rpm (> 1000 rpm).
- Akurasi : $\pm (0,05\% + 1 \text{ digit})$.
- Waktu Sampling : 0,8 sec.
- Memori : Nilai Terakhir, Nilai Tertinggi,
Nilai Terendah.
- Jarak Ukur : 50-200 mm LED, 50-500 mm Laser.
- Basis Waktu : Quartz Crystal.
- Rangkaian : *Exclusive One-Chip of Microcomputer LSI Circuit.*
- Baterai : 6FF22 9 Volt.
- Konsumsi Daya : $\pm 35 \text{ mA LED}$, $\pm 30 \text{ mA Laser}$.
- Suhu Operasi : $0-50^{\circ} \text{ C}$.
- Dimensi : $131 \times 70 \times 29 \text{ mm}$
- Berat : 160 g (termasuk baterai).

2.8. Real Time Clock

Real-timeClock (RTC) adalah komponen digunakan pada rangkaian elektronik untuk menjaga waktu relatif sesuai dengan waktu nyata. RTC pada umumnya dihubungkan dengan rangkaian mikrokontroler dengan komunikasi SPI atau I2C serial bus.

Modul RTC DS1307 memiliki ketelitian *error* sebesar 1 menit per tahun.



Gambar 2.15 Modul RTC DS1307.

Berikut fungsi pin dari Modul RTC DS1307:

- Pin Vcc : Sumber energi utama, tegangan kerja 5 volt.
- Pin GND : *Ground*.
- Pin SCL : Saluran *Clock* untuk komunikasi I2C.
- Pin SDA : Saluran Data untuk komunikasi I2C.
- Pin BAT : Saluran baterai eksternal.

2.9. *BuckConverter*

Step-downswitchingRegulators yang juga disebut *buck converters* adalah penurunan level tegangan yang nilai keluaran dapat diatur. Saat ini, *step-downswitchingregulators and controllers* tersedia dengan arsitektur internal yang telah dioptimalkan untuk aplikasi yang lebih spesifik, seperti penghemat daya, efisiensi tinggi, atau rendah biaya.

Mini 360 DC-DC *Buck Converter* adalah modul *buck converterstep down* yang menggunakan integral power induktor dan chip kontrol penyearah sinkron.

Berikut spesifikasi dari DC-DC Buck Converter:

- IC Buck Regulator : MPS MP2307DN
- Tegangan *Input* : 4,75-23 VDC
- Tegangan *Output* : 1,0-17 VDC
- Arus Lonjakan : 3A
- Arus Continu : 1,8 A
- Efisiensi Max. Konversi : 95% (5 Vin, 3,3 Vout)
- *Swthching Frequency* : 340 kHz
- Riak Keluaran tanpa Beban : 30 mV
- Beban Regulator : 0,5 %
- Tegangan Regulator : 2,5 %



Gambar 2.16 Mini 360 DC-DC *Buck Converter*

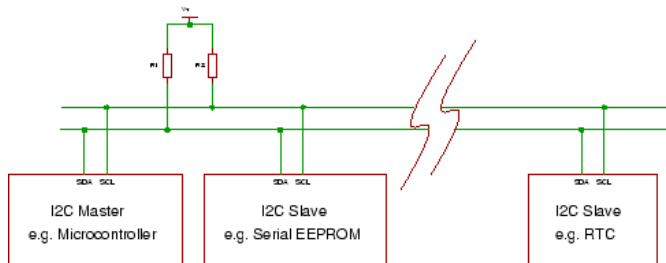
- Jangkauan Suhu Operasi : -40 sampai +85° C
- Ukuran : 17 × 11 × 3,8 mm

2.10. Komunikasi I²C

I²C (*Inter-Integrated Circuit*), pertama kali dirancang oleh perusahaan Philips Semiconductors. Metode komunikasi ini memungkinkan komunikasi antar *device* yang dilengkapi dengan protokol I²C hanya dengan dua kabel. Pengiriman/penerimaan data/informasi secara serial dengan menggunakan data *line*, *Serial Data* (SDA), dan sebuah sumber *clock*, *Serial Clock* (SCL). Kedua jalur, jalur data, SDA, dan jalur *clock*, SCL, harus di-*Pull-UP* dengan resistor eksternal. Gambar 2.17 memperlihatkan skema arsitektur I²C dengan satu *master*, mikrokontroler, dan beberapa *slave*, Serial EEPROM dan RTC.^[8]

Master adalah perangkat yang memiliki kontrol pada BUS, master mengontrol sinyal *serial clock*, sinyal START, dan sinyal STOP. *Master* memiliki kendali untuk menentukan komunikasi dimulai atau diakhiri. Sementara *slave* adalah perangkat yang menunggu perintah dari *master*, menulis/menerima data atau mengirim data ke *master*.

Pada komunikasi I²C, meski *serial clock* hanya berasal dari *master*, namun *slaved* dapat menahan sinyal *serial clock*, apabila *slave* belum siap menerima data, sinyal *serial clock* akan ditahan pada kondisi “LOW”. Hal ini biasa terjadi pada peralatan yang lambat.

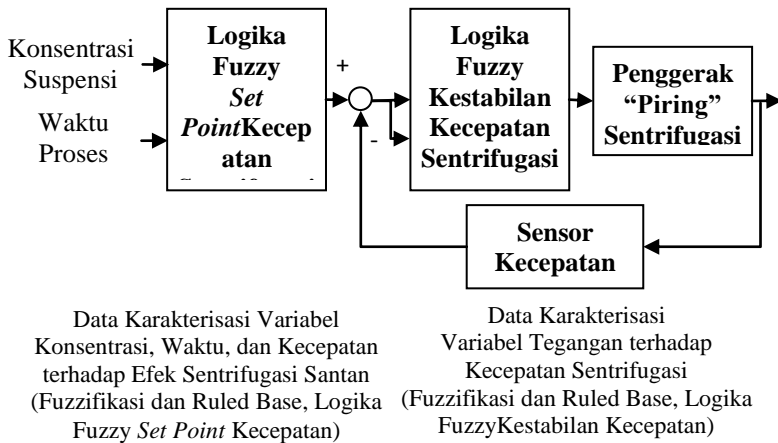


Gambar 2.17 Skema komunikasi I²C

.....*Halaman ini sengaja dikosongkan*.....

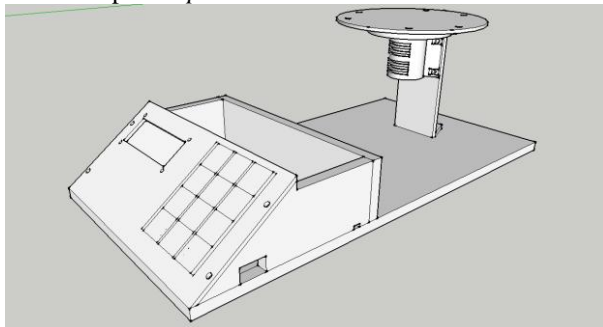
BAB III

PERANCANGAN ALAT SENTRIFUGASI

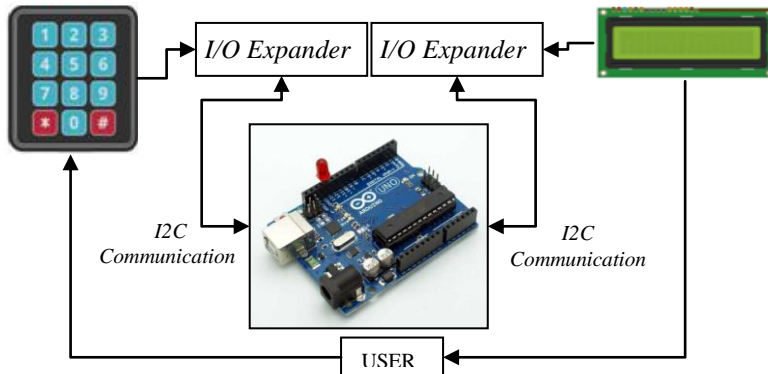


Gambar 3.1 Diagram blok sistem pengendali kecepatan alat sentrifugasi

Alat sentrifugasi yang dirancang pada tugas akhir ini dilengkapi dengan sistem kontroler logika fuzzy. Konsentrasi dan lama waktu proses sentrifugasi dari suspensi menjadi masukan logika fuzzy kecepatan *set point* sentrifugasi. Nilai kecepatan *set point* dari logika fuzzy pertama akan diakusisi oleh logika fuzzy kedua, nilai *set point* akan menjadi nilai parameter untuk kecepatan putar aktual alat sentrifugasi yang diperoleh dari akusisi data tachometer encoder. Perbedaan nilai dari kecepatan *set point* dan kecepatan aktual, *error*, dan perubahan perbedaan keduanya, *deltaError*, akan menjadi masukan bagi logika fuzzy kedua akan akan diproses untuk mendapatkan nilai PWM yang tepat untuk mencapai *set point*.^{[3]-[5]}



Gambar 3.2 Desain alat sentrifugasi.



Gambar 3.3 Skema sistem antarmuka alat sentrifugasi.

Perancangan Alat sentrifugasi ini dibagai menjadi dua kategori, Perancangan Perangkat Keras dan Perancangan Perangkat Lunak.

3.1. Perancangan Perangkat Keras

Alat sentrifugasi ini disusun atas data tiga sistem utama, ketiga sistem tersebut diantaranya: sistem antarmuka, sistem pewaktu pada alat sentrifugasi, sistem pemutar/penggerak pada alat sentrifugasi, dan sistem sensor kecepatan sentrifugasi

3.1.1. Sistem Antarmuka

Gambar 3.3 adalah skema sistem antarmuka pada alat sentrifugasi, disusun dari *keypad* 4x3 sebagai perangkat masukan, arduino uno sebagai perangkat mikrokontroler, dan LCD 16x2 sebagai perangkat *display*. Komunikasi antar perangkat menggunakan metode komunikasi *i2c*. Metode komunikasi *i2c* dipilih dikarenakan keterbatasan *port I/O* pada Arduino Uno.

3.1.2. Sistem Pewaktu pada Alat Sentrifugasi

Penggunaan *Real Time Clock* (RTC) pada alat sentrifugasi dikarenakan untuk meningkatkan ketelitian atau ketepatan waktu proses sentrifugasi.



Gambar 3.4 Skema Sistem Pewaktu Alat Sentrifugasi

Pewaktu tidak aktif pada saat alat sentrifugasi baru dinyalakan, pewaktu baru aktif ketika nilai massa air “di-OK”. Dengan ditekannya tombol “OK” maka pewaktu proses aktif, menjadi salah satu masukan dari proses logika fuzzy *Set Point* kecepatan sentrifugasi dan pembatas waktu proses sentrifugasi. Dengan batas waktu 30 menit, apabila data waktu terlama dari karakterisasi lebih cepat dari 30 menit maka waktu tersebutlah yang digunakan sebagai waktu batas sentrifugasi.

3.1.3. Sistem Penggerak/Pemutar pada Alat Sentrifugasi

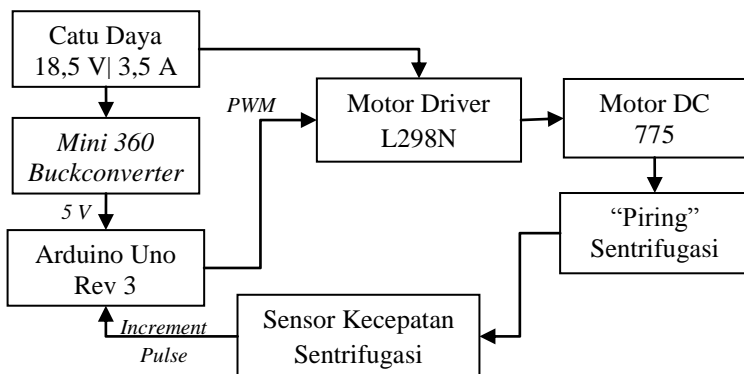
Sistem penggerak alat sentrifugasi terdiri dari beberapa perangkat, diantaranya: Catu daya, Mini 360 Buck Converter, Arduino Uno, Modul Motor Driver L298N, Motor DC 775, “Piring” Sentrifugasi, dan Sensor Kecepatan.

Catu daya yang digunakan pada alat sentrifugasi ini memiliki tegangan keluaran 18,5 volt dengan arus 3 A. Penggunaan catu daya ini untuk memenuhi kebutuhan arus pada motor dc 775 untuk memutar suspensi santan. Semakin besar torsi motor semakin besar arus yang dibutuhkan motor.

Mini 360 buck converter digunakan untuk menurunkan tegangan keluaran catu daya, 18 volt, yang digunakan untuk men-supply Arduino Uno, 5 volt.

Arduino Uno digunakan sebagai mikrokontroler, perangkat yang mengeluarkan sinyal PWM untuk men-drive kecepatan Motor DC 775.

Modul Motor Driver L298N digunakan untuk sebagai perangkat motor drive berdasarkan sinyal PWM keluaran Arduino Uno.



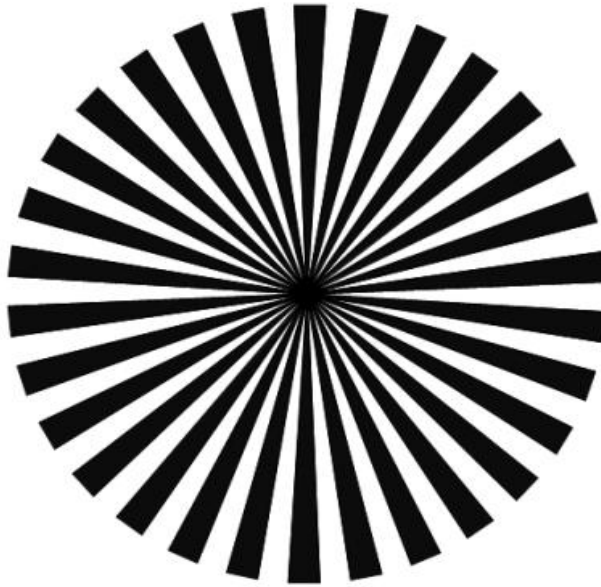
Gambar 3.5 Skema Sistem Pemutar pada Alat Sentrifugasi

Perangkat “piring” sentrifugasi dan sensor kecepatan sentrifugasi akan dibahas pada bagian Sensor Kecepatan Sentrifugasi.

Sensor kecepatan sentrifugasi pada tugas akhir ini dirancang berbasis sensor *increment encoder*. Perangkat yang menyusun subsistem Sensor Kecepatan Sentrifugasi diantaranya: *Sensor Increment Encoder*, “Piring” Sentrifugasi, dan Arduino Uno.

[illegible]

30



Gambar 3.7 Perancangan “piring” sentrifugasi

Penggunaan inframerah sebagai sumber cahaya dinilai lebih baik untuk metode pantulan, dikarenakan cahaya dari LED biasa bersifat tidak terfokus dan nampak oleh penglihatan, sehingga hal itu akan mengurangi nilai estetika dari alat sentrifugasi. Sementara cahaya dari LED Inframerah walau bersifat menyebar, namun karena inframerah tidak nampak oleh penglihatan manusia, sehingga tidak mempengaruhi nilai estetika dari alat sentrifugasi. Dan cahaya pantulan tetap bisa diterima photodiode.

“Piring” Sentrifugasi dirancang agar mampu menyerap dan memantulkan cahaya inframerah yang dipancarkan oleh LED Inframerah. Warna kontras Hitam-Putih dipilih agar hasil sinyal keluaran sensor *Increment Encoder* jelas perbedaannya. Terdapat 30 garis warna hitam dan 30 garis warna putih dengan lebar sudut sama besar, ± 6 derajat.

	M	a	s	s	a		S	u	s	p	e	n	s	i			
														g	r	a	m

Gambar 3.8 Skema tampilan menu awal

3.2. Perancangan Perangkat Lunak

Tabel 3.1 Skema fungsi *keypad* pada setiap tahap

No.	Keypad 3x4	Tahap1	Tahap 2	Tahap 3	Tahap 4
1	1	1	1	-	-
2	2	2	2	-	-
3	3	3	3	-	-
4	4	4	4	-	-
5	5	5	5	-	-
6	6	6	6	-	-
7	7	7	7	-	-
8	8	8	8	-	-
9	9	9	9	-	-
10	0	0	0	-	-
11	*	<i>Cancel</i>	<i>Cancel</i>	-	-
12	#	Ya/OK	Ya/OK	-	-
ket:		-			
Tahap 1:		Memasukkan Nilai Massa Suspensi Santan			
Tahap 2:		Memasukkan Nilai Massa Air			
Tahap 3:		Proses Sentrifugasi Berjalan			
Tahap 4:		Proses Sentrifugasi Berakhir			

Alat sentrifugasi pada penelitian ini dirancang secara keseluruhan tersusun atas empat sistem perangkat lunak, diantaranya: Sistem Antarmuka, Sistem Logika Fuzzy *Set Point*, Sistem Sensor Kecepatan dan Sistem Logika Fuzzy Kestabilan.

Tahap proses pemisahan suspensi/larutan alat sentrifugasi ini dibagi menjadi empat tahap, diantaranya: tahap memasukkan nilai massa santan, tahap memasukkan nilai massa air, tahap proses sentrifugasi, dan tahap proses selesai.

3.2.1. Sistem Antarmuka

Pada tabel 3.1 sepuluh angka yang terdapat pada keypad 3x4 difungsikan sebagai input angka. Sementara dua pin lain, “*” dan “#”, difungsikan sebagai input instruksi.

M	a	s	s	a		A	i	r							
													g	r	a
													m		

Gambar 3.9 Skema tampilan menu ke-dua.

r	p	m				S	e	t			A	k	t	
						x	x	x	x	x		x	x	x

Gambar 3.10 Tampilan menu pada tahap proses sentrifugasi.

Gambar 3.8 adalah tampilan menu awal sekaligus menu untuk memasukkan berapa berat suspensi yang akan dipisahkan.

Tombol “YA” untuk menuju menu berikutnya, Gambar 3.9, sementara tombol “Cancel” untuk membatalkan bila terjadi kesalahan saat memasukkan nilai berat suspensi, Tabel 3.1.

Gambar 3.9 adalah tampilan menu untuk memasukkan berapa berat air yang ditambahkan. Besar nilai konsentrasi suspensi mengacu pada persamaan 3.1. Dengan menekan “OK” maka **Pewaktu Proses** menjadi “ON” dan sistem sentrifugasi menuju tahap berikutnya.

$$\text{Konsentrasi Suspensi} = \frac{\text{Berat Suspensi}}{\text{Berat Suspensi} + \text{Berat Tambahan Air}} \quad (3.1)$$

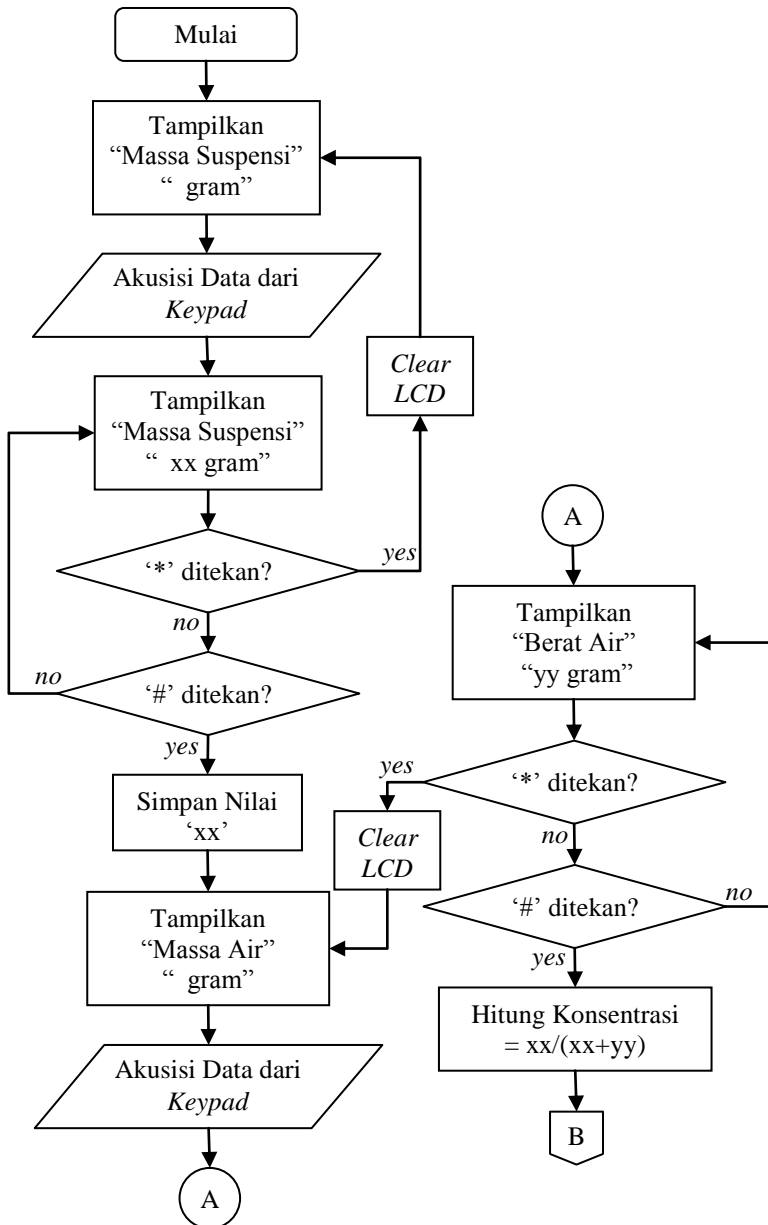
Setelah mendapatkan nilai konsentrasi suspensi dan nilai waktu proses maka proses sentrifugasi santan dimulai. Sistem logika fuzzy *set point* akan menentukan kecepatan sentrifugasi, sistem sensor kecepatan sentrifugasi akan melakukan pembacaan kecepatan sentrifugasi berdasarkan jumlah *increment* dalam setiap periode perulangan sistem. Kedua hasil kecepatan itu akan ditampilkan pada LCD, gambar 3.10.

Waktu proses yang merupakan masukan sistem pengendali kecepatan, maka nilai waktu proses akan terus di-*feedback*-kan kembali ke proses sentrifugasi pada setiap periode perulangan sistem. Nilai waktu proses merupakan indikator kondisi suspensi santan. Perubahan nilai waktu proses akan membuat nilai *set point* kecepatan sentrifugasi akan berubah sesuai keputusan logika fuzzy *set point* kecepatan sentrifugasi. Proses sentrifugasi akan berhenti ketika waktu proses mencapai 30 menit

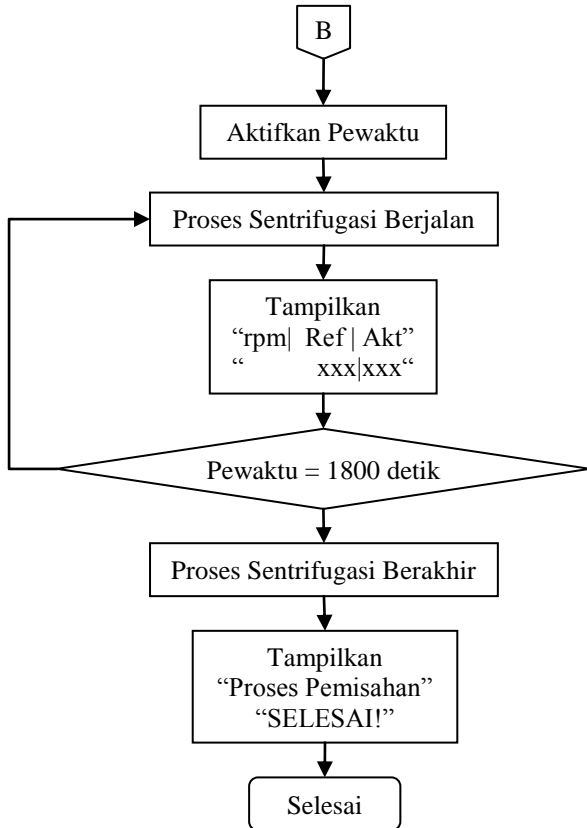
P	r	o	s	e	s		P	e	m	i	s	a	h	a	n
				S	E	L	E	S	A	I	!				

Gambar 3.11 Tampilan pada tahap terakhir.

Gambar 3.12a dan 3.12b merupakan skema diagram alir antar-muka alat sentrifugasi:

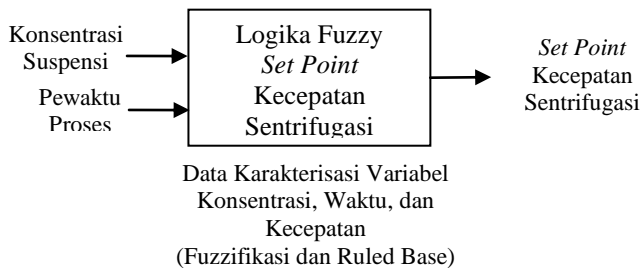


Gambar 3.12a Diagram alir skema sistem antarmuka

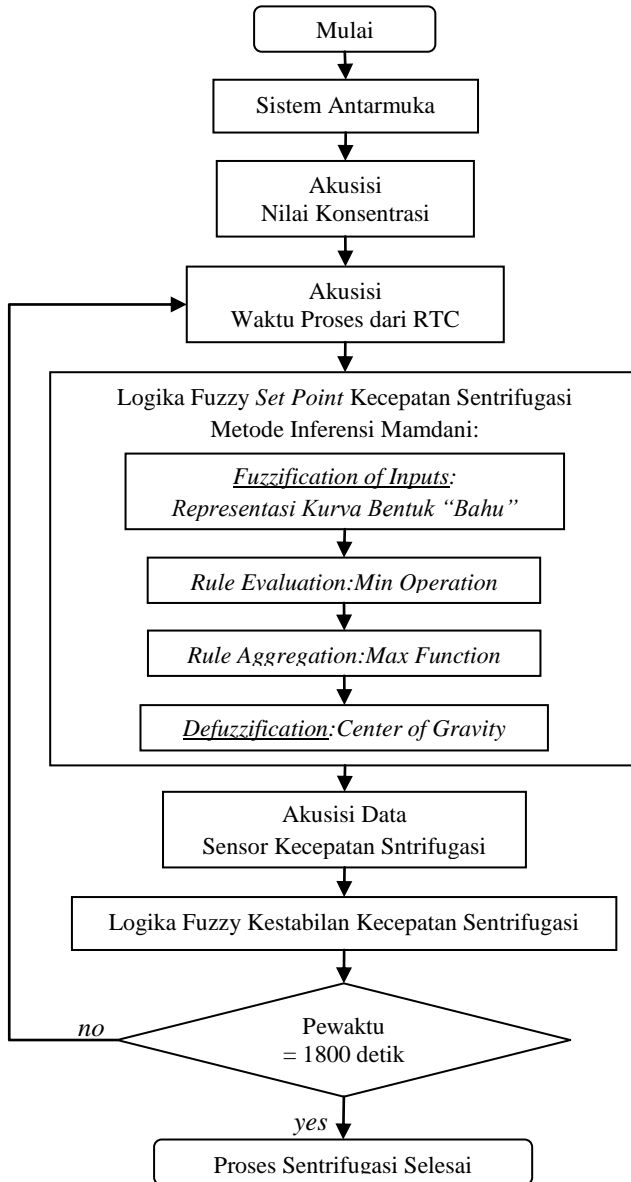


Gambar 3.12bDiagram alir skema sistem antarmuka

3.2.2. Logika Fuzzy *Set Point* Kecepatan Sentrifugasi



Gambar 3.13 Diagram blok sistem logika fuzzy *set point* kecepatan.

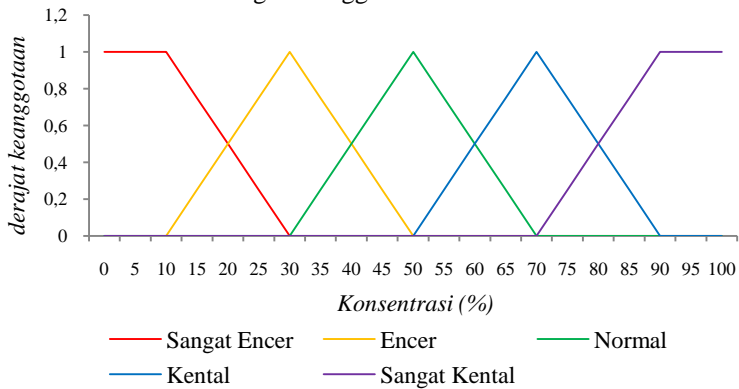


Gambar 3.14 Diagram alir logika fuzzy *set point* kecepatan sentrifugasi.

Data karakterisasi variabel konsentrasi, waktu , dan kecepatan menjadi acuan dalam fuzzifikasi dan *ruled base* dari logika fuzzy *set point* kecepatan sentrifugasi. Variabel konsentrasi dan variabel waktu menjadi variabel masukan untuk menentukan variabel kecepatan sentrifugasi, variabel keluaran.Selama proses sentrifugasi *keypad* dinonaktifkan.

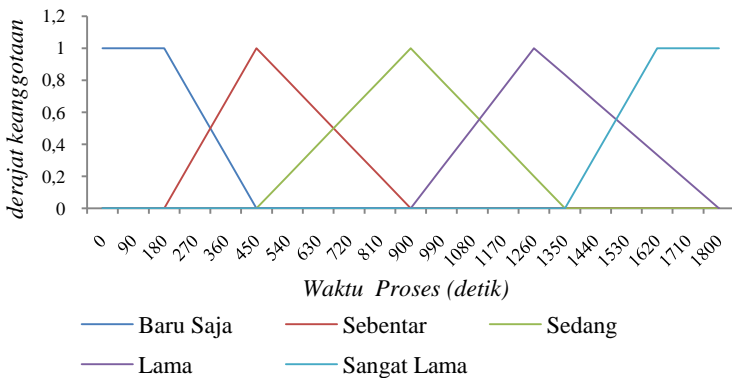
Representansi fungsi keanggotaan yang digunakan untuk fuzzifikasi seluruh variabel masukan, variabel konsentrasi dan variabel proses waktu, adalah representasi bentuk “bahu”. Pemilihan representasi bentuk bahu dikarenakan kemudahan pengaplikasian dalam program mikrokontroler.^[5]

Berikut skema fungsi keanggotaan variabel konsentrasi:



Gambar 3.15 Grafik keanggotaan variabel konsentrasi.

Berikut skema fungsi keanggotaan variabel waktu proses:



Gambar 3.16 Grafik kenggotaan variabel waktu proses.

Tabel 3.2*Ruled Base Output, Set Point*Kecepatan Sentrifugasi

Waktu Proses	Baru	Sebentar	Sedang	Lama	Sangat Lama
Konsentrasi	Saja				
Sangat Rendah	400	200	200	0	0
Rendah	400	400	200	200	0
Normal	600	400	400	200	0
Tinggi	800	600	400	400	0
Sangat Tinggi	800	800	600	400	0

Metode **mamdani** adalah metode inferensi yang digunakan pada logika fuzzy *set point* kecepatan sentrifugasi. Metode *Min Operation* adalah *rule evaluation* yang digunakan untuk menentukan besar nilai derajat keanggotaan dari kedua variabel masukan disetiap periode perulangan sistem.

Tabel 3.2 adalah *ruled base ouput* “asumsi” logika fuzzy *set point* kecepatan sentrifugasi. Metode *max function* adalah *rule aggregation* yang digunakan untuk menentukan besar *output set point* kecepatan dari *Ruled Base* yang telah dirancang. *Ruled base* akan disesuaikan kembali dengan data karakterisasi suspensi santan.

Metode *centre of gravity* (persamaan 2.20) digunakan untuk defuzzifikasi dalam menentukan *set point* kecepatan sentrifugasi yang dibutuhkan alat sentrifugasi untuk memisahkan suspensi disetiap periode perulang sistem.

3.2.3. Sensor Kecepatan Sentrifugasi

Jumlah pulsa *increment* dalam satu putaran “piring” sentrifugasi adalah 30 bernilai “HIGH” dan 30 pulsa bernilai “LOW”. Maka didapatkan kecepatan putar “piring” sentrifugasi dapat adalah:

$$Kecepatan_{Angular} = Frekuensi\ Keluaran \times 2 \text{ (rpm)} \quad (3.1)$$

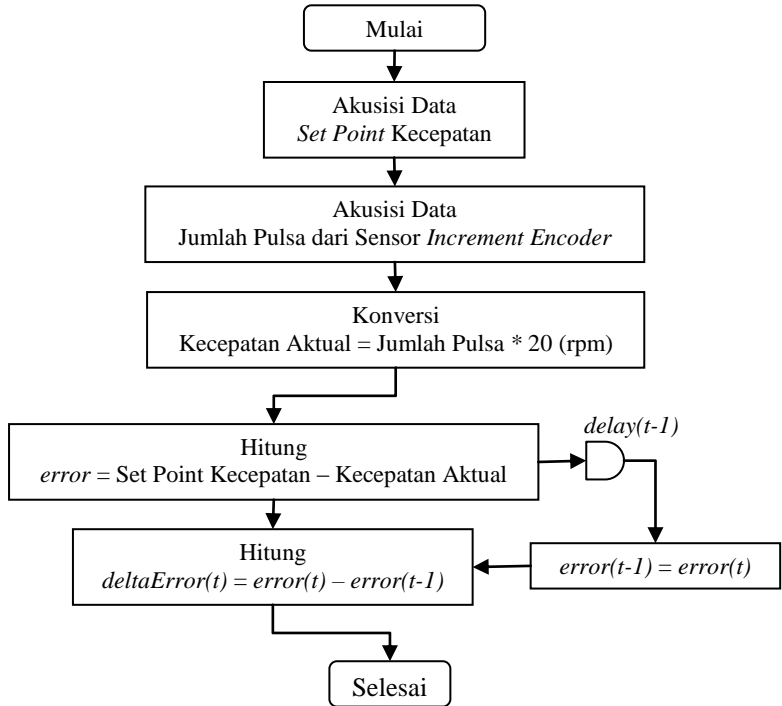
Dengan setiap periode perulangan sistem proses adalah 0,1 detik atau 10 Hz. Besar kecepatan sentrifugasi dalam setiap perulangan sistem adalah:

$$Kecepatan_{sentrifugasi} = Jumlah\ Pulsa \times 20 \text{ (rpm)} \quad (3.2)$$

Variabel yang masuk pada Logika Fuzzy Kestabilan Kecepatan Sentrifugasi adalah *error* dan *deltaError*. Berikut persamaan keduanya:

$$error(t) = Kecepatan\ Set\ Point(t) - Kecepatan\ Aktual(t) \quad (3.3)$$

$$deltaError(t) = Error(t) - Error(t - 1) \quad (3.4)$$



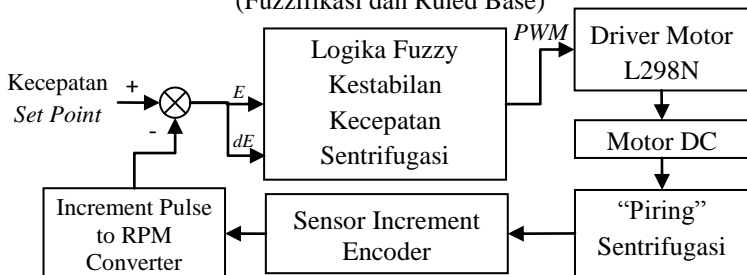
Gambar 3.17 Diagram alir sensor kecepatan sentrifugasi.

3.2.4. Logika Fuzzy Kestabilan Kecepatan Sentrifugasi

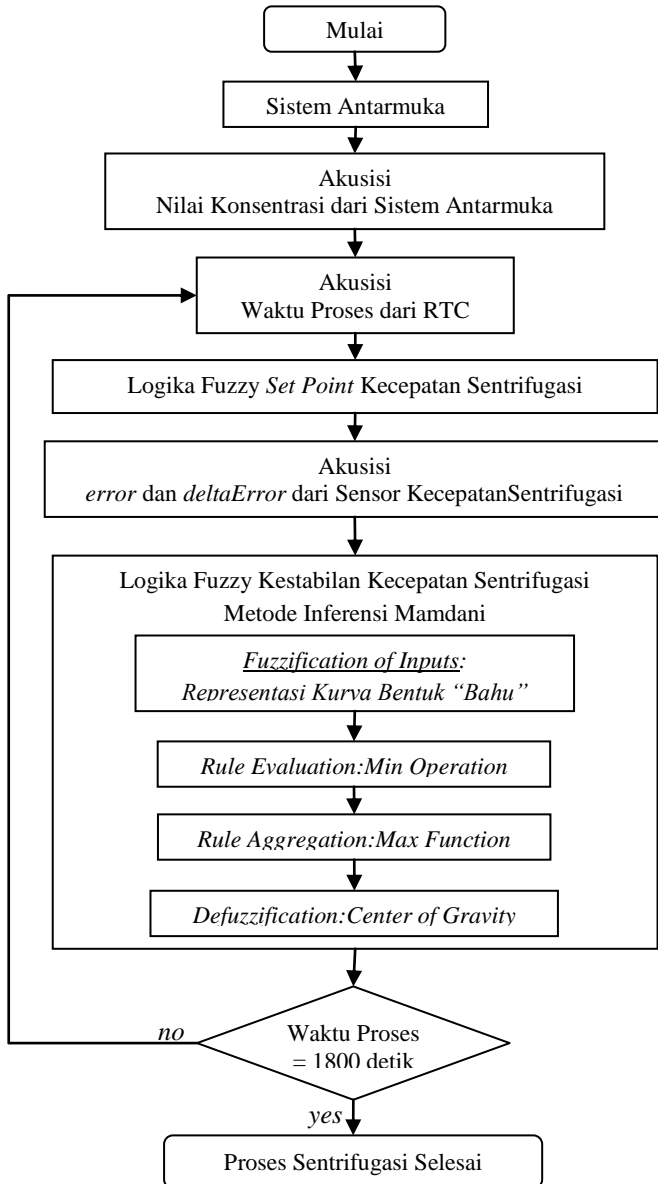
Data Karakterisasi Kecepatan

Motor DC

(Fuzzifikasi dan Ruled Base)



Gambar 3.18 Skemadiagram blok logika fuzzy kestabilan kecepatan sentrifugasi.



Gambar 3.19 Diagram alir logika fuzzy kestabilan kecepatan sentrifugasi.

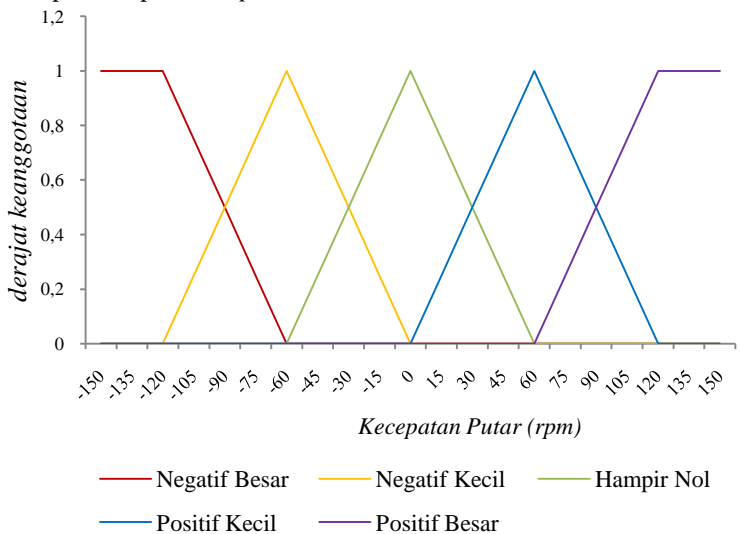
Data karakterisasi kecepatan Motor DC terhadap tegangan menjadi acuan dalam fuzzifikasi dan *ruled base* logika fuzzy kestabilan kecepatan sentrifugasi.

Hasil keputusan logika fuzzy *set point* kecepatan sentrifugasi menjadi variabel parameter pada Kontroler Logika Fuzzy Kestabilan Kecepatan Sentrifugasi yang harus dipenuhi.

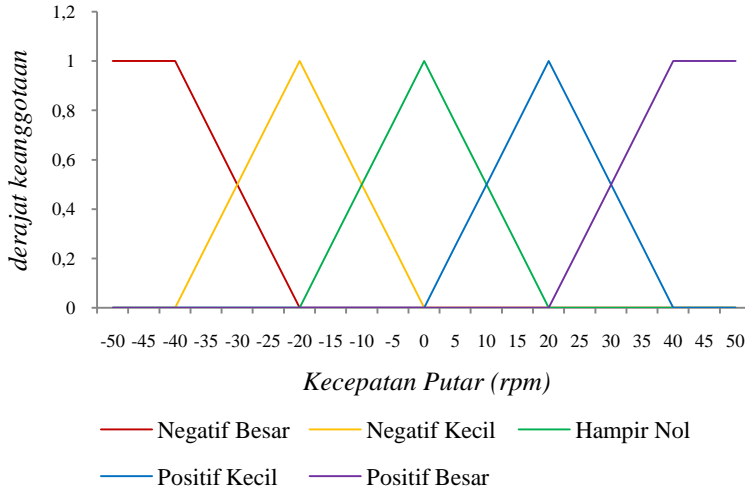
Variabel *input* pada Logika Fuzzy Kestabilan Kecepatan Sentrifugasi adalah *errordan deltaError*. Nilai *error* diperoleh dari perbedaan nilai kecepatan *set point* dengan kecepatan aktual. Sementara nilai *deltaError* diperoleh dari perbedaan *error* periode sistem saat ini dengan *error* periode sistem sebelumnya.

Representasi kurva bentuk “bahu” digunakan untuk fuzzifikasi *Membership function* setiap nilai variabel masukan, *error* dan *deltaError*. Berikut skema *membership function* variabel *input* pada logika fuzzy kestabilan kecepatan sentrifugasi:

Untuk metode inferensi yang digunakan pada Logika Fuzzy Kestabilan Sentrifugasi adalah metode *mamdani*. Metode *minoperation* pada *rule evaluation* yang digunakan untuk menentukan nilai keanggotaan dari kedua variabel masukan, metode *max function* pada *rule aggregation* dan metode defuzzifikasi *centre of area* untuk menentukan *output* koreksi *duty cycle* dari PWM agar kecepatan aktual sentrifugasi mencapai kecepatan *set point*.



Gambar 3.20 Grafik keanggotaan variabel *error*.



Gambar 3.21 Grafik keanggotaan variabel *deltaError*.

Tabel 3.3 *Ruled base output koreksi duty cycle PWM.*

Error	Negatif	Negatif	Hampir	Positif	Positif
<i>deltaError</i>	Besar	Kecil	Nol	Kecil	Besar
Negatif Besar	-6	-3	-3	0	3
Negatif Kecil	-6	-3	0	0	6
Hampir Nol	-6	0	0	0	6
Positif Kecil	-6	0	0	3	6
Positif Besar	-3	0	3	3	6

Hasil keputusan logika fuzzy kestabilan kecepatan sentrifugasi akan menentukan nilai koreksi *duty cycle* PWM yang dihubungkan kepin enable B L298N, gambar 2.12. Secara tidak langsung nilai keputusan logika fuzzy digunakan sebagai koreksi tegangan relatif pada aktuatur, Motor DC.

$$V_{Relatif} = V_{Sumber} \cdot Duty\ Cycle \quad (3.5)$$

$$Duty\ Cycle = \frac{Time\ High}{Time\ High + Time\ Low} \% \quad (3.6)$$

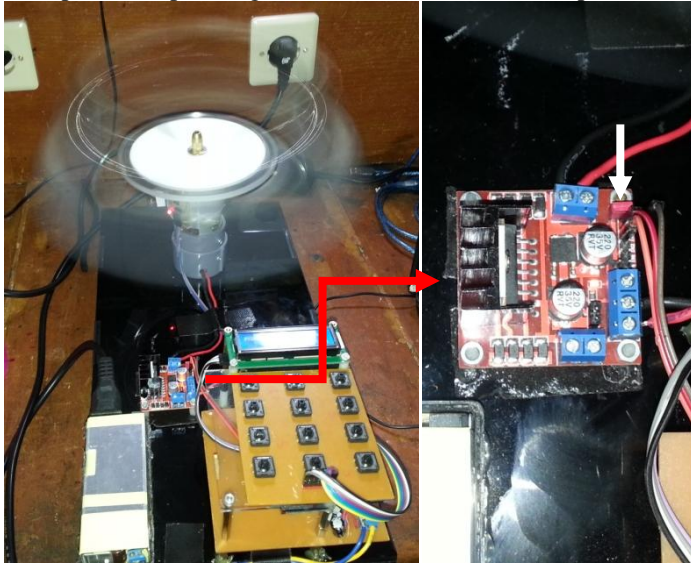
BAB IV

UJI COBA DAN ANALISIS

4.1. Pengujian Kecepatan Maksimal Alat Sentrifugasi

Pengujian dilakukan dengan menyalakan alat sentrifugasi dan menghubungkan pin enable B dengan *pin jumper* (panah putih). Dengan menghubungkan *pin enable* B maka motor DC akan berputar dengan kecepatan penuh. Pengukuran dilakukan dengan menggunakan tachometer digital tipe DT-2234C⁺.

Dari data yang tertera pada Tabel 4.1 menunjukkan hasil pembacaan pada Tachometer DT-2234C⁺ bahwa kecepatan putar maksimum alat sentrifugasi pada kondisi tanpa beban berkisar pada kecepatan 1129 sampai 1139 rpm dengan rata-rata sebesar 1134,25 rpm.



Gambar 4.1 Pengujian kecepatan maksimal.

Tabel 4.1 Kecepatan maksimal alat sentrifugasi tanpa beban.

No.	Pengambilan Ke-	DT-2234C ⁺ , (rpm)
1	1	1139
2	2	1137
3	3	1132
4	4	1129
Rata-rata		1134,25

Tabel 4.2 Kecepatan maksimal alat sentrifugasi berbeban.

No.	Pengambilan Ke-	DT-2234C ⁺ , (rpm)
1	1	926
2	2	931
3	3	936
4	4	934
Rata-rata		931,75

Tabel 4.2 menunjukkan hasil pengujian kecepatan maksimal alat sentrifugasi pada kondisi berbeban, beban, tiga botol suspensi, yang diberikan pada alat sentrifugasi seberat 190 gram. Hasil pembacaan pada Tachometer DT-2234C⁺ bahwa kecepatan putar maksimal alat sentrifugasi berkisar pada kecepatan 926 sampai 936 rpm dengan rata-rata sebesar 931,75 rpm.

Dari dua hasil pengujian kecepatan maksimal alat sentrifugasi, dengan mengambil kecepatan terbaca pada tachometer DT-2234C⁺ sebagai kecepatan aktual menunjukkan bahwa, pembebanan pada alat sentrifugasi mempengaruhi kecepatan sentrifugasi, rata-rata kecepatan sentrifugasi aktual tanpa beban 1134,25 rpm sementara rata-rata kecepatan sentrifugasi aktual saat berbeban 931,75 rpm. Setelah pembebanan kecepatan sentrifugasi turun 202,5 rpm atau 17,85% dari kecepatan tanpa beban.

4.2. Pengujian Sensor Kecepatan Sentrifugasi

Pengujian dilakukan dengan melihat sinyal keluaran sensor *increment encoder* terhadap kecepatan putar “piring” sentrifugasi pada tegangan tertentu, sinyal keluaran sensor encoder tampilkan pada osiloskop untuk melihat bentuk dan frekuensi pulsa yang dikeluarkan sensor *increment encoder*.

Tabel 4.3 Hasil uji sensor *increment encoder*.

No.	Tegangan (Volt)	Sensor Encoder		DT-2234C ⁺
		Frekuensi (Hz)	Kecepatan Putar = $f \times 2$, (rpm)	(rpm)
1	3	173	346	356
2	6	290	580	509
3	9	403	806	819
4	12	480	960	987
5	15	568	1136	1135
6	18	595	1190	1258



Gambar 4.2 Pengujian sensor *increment encoder*.

Dari data yang tertera pada tabel 4.3 menunjukkan bahwa perubahan jumlah pulsa setiap detik yang dikeluarkan pada sensor *increment encoder* berbanding lurus dengan perubahan tegangan. Terdapat perbedaan hasil pembacaan antara sensor *increment encoder* dengan DT-2234C⁺, dari enam pengujian rata-rata perbedaan sebesar 31,67 rpm, perbedaan tertinggi terdapat pada tegangan 6 volt, mencapai 71 rpm, perbedaan terkecil terdapat pada tegangan 15 volt, hanya 1 rpm.

4.3. Pengujian *Duty Cycle* Terhadap Sistem Penggerak Alat Sentrifugasi

Dilakukan pengujian pada sistem penggerak alat sentrifugasi. Pengujian ini bertujuan untuk melihat respon sistem penggerak alat sentrifugasi terhadap perubahan *duty cycle* PWM.

Dari tabel 4.4 menunjukkan bahwa kekelitiran sensor kecepatan pada sistem penggerak alat sentrifugasi cenderung mengalami penurunan ketika *Duty Cycle* PWM semakin tinggi. Nilai ketelitian tertinggi sensor kecepatan sentrifugasi terjadi pada *duty cycle* PWM, 10%, *error* sebesar 1,52. Nilai ketelitian terendah terjadi pada *duty cycle*, PWM 90%, *error* sebesar 13,31. Pada pembangkitan *duty cycle* 100% Arduino tidak mengirimkan data kecepatan.

Tabel 4.4 Respon sistem penggerak alat sentrifugasi.

No.	Duty Cycle (%)	Kecepatan (rpm)		error (%)
		Sensor Kecepatan	DT-2234C	
1	10	266	262	1,52
2	20	431,0175	415	3,85
3	30	544,4385	510	6,75
4	40	628,0350	609	3,12
5	50	705,3508	675	4,49
6	60	761,1052	730	4,26
7	70	835,7894	775	7,84
8	80	889,4385	812	9,53
9	90	951,8596	840	13,31

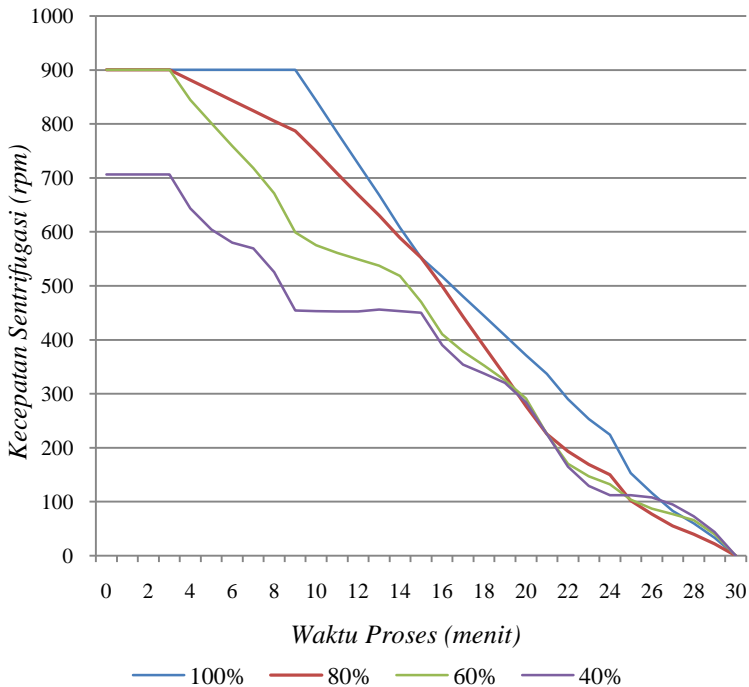
4.4. Pengujian Logika FuzzySet PointKecepatan Sentrifugasi

Dilakukan pengujian pada logika fuzzy kecepatan *set point* sentrifugasi. Pengujian bertujuan untuk melihat respon sistemlogika fuzzy dalam menentukan kecepatan *set point*terhadap beberapa level konsentrasi dengan asumsi waktu proses selama 30 menit.

Dari gambar 4.2 menunjukkan bahwa logika fuzzy *set point* kecepatan sentrifugasimampu mengatur kecepatansentrifugasi sesuai dengan perubahan variabel waktu proses di setiap konsentrasi santan.



Gambar 4.3 Pengujian *set point* kecepatan sentrifugasi



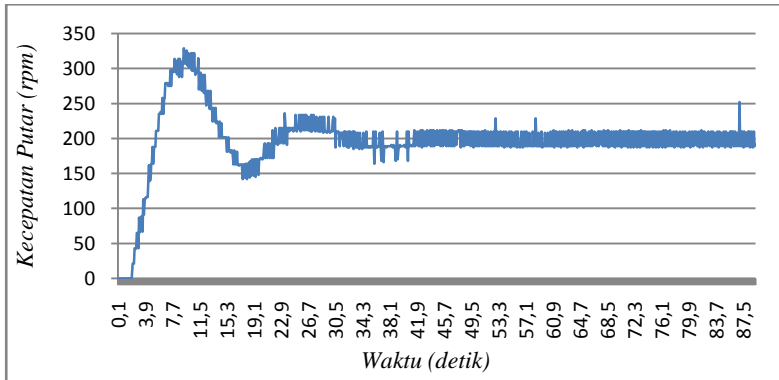
Gambar 4.4 Respon logika fuzzy *set point* kecepatan sentrifugasi terhadap perubahan waktu.

4.5. Pengujian Logika Fuzzy Kestabilan Kecepatan Sentrifugasi

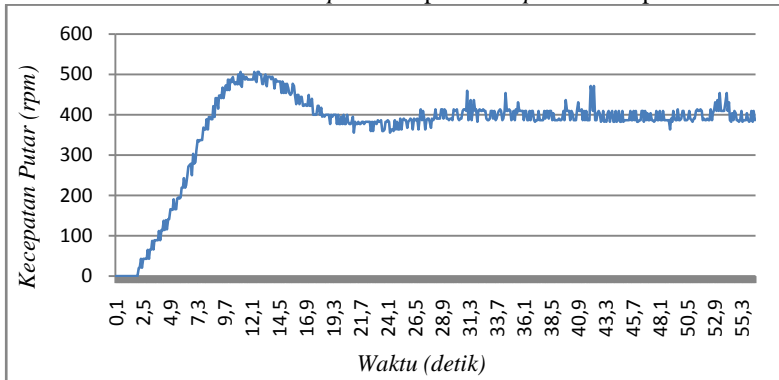
Pada pengujian ini kontroler logika fuzzy akan mendapat masukan variabel parameter berupa kecepatan *set point* dan variabel *feedback* berupa kecepatan aktual. Variabel kecepatan *set point* bersifat tetap akan ditanamkan ke dalam program, sementara kecepatan aktual didapatkan dari akusisi data dari sensor kecepatan sentrifugasi.

4.5.1. Pengujian Tanpa Beban

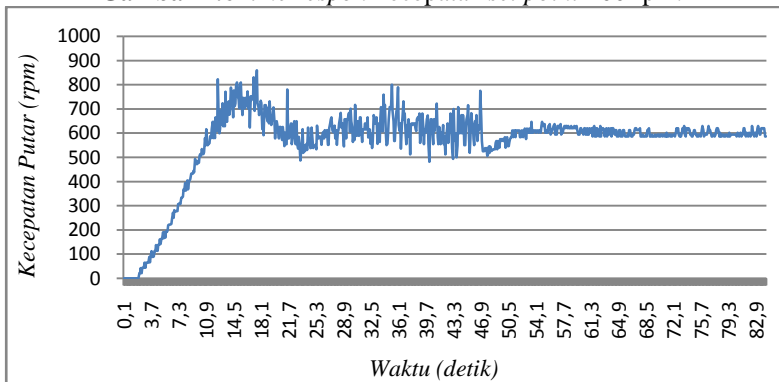
Pengujian pada alat sentrifugasi dengan kontroler dilakukan dengan memberikan beberapa nilai pada variabel parameter, kecepatan *Set Point*. Berikut grafik *time respon* kontroler terhadap beberapa nilai kecepatan *Set Point*, tanpa beban.



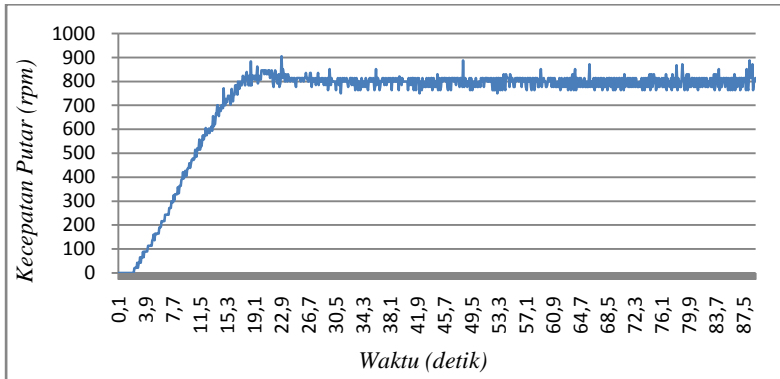
Gambar 4.5 *Time respon kecepatan set point 200 rpm.*



Gambar 4.6 *Time respon kecepatan set point 400 rpm.*



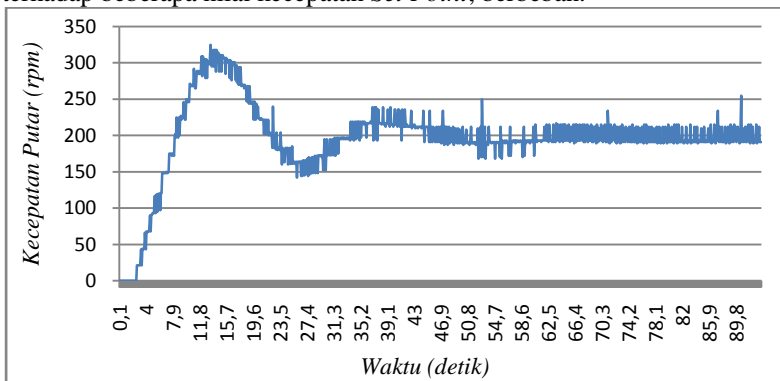
Gambar 4.7 *Time respon kecepatan set point 600 rpm.*



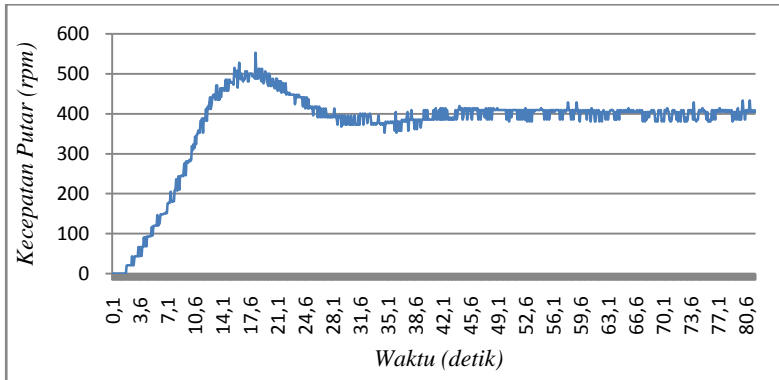
Gambar 4.8 *Time respon* kecepatan set point 800 rpm.

4.5.2. Pengujian Berbeban

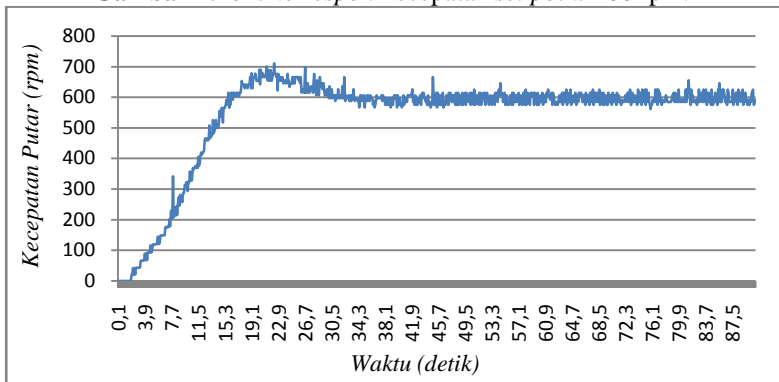
Perlakuan pada pengujian sama dengan pengujian tanpa beban, hanya yang membedakan pada pengujian ini adalah adanya beban yang harus diputar oleh alat sentrifugasi. Berikut grafik *time respon* kontroler terhadap beberapa nilai kecepatan Set Point, berbeban.



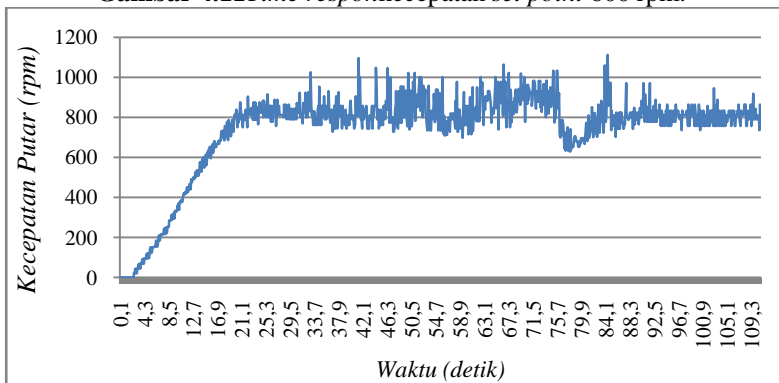
Gambar 4.9 *Time respon* kecepatan set point 200 rpm.



Gambar 4.10Time respon kecepatan set point 400 rpm.



Gambar 4.11Time responkecepatan set point 600 rpm.



Gambar 4.12Time responkecepatan set point 800 rpm.

Tabel 4.5 Spesifikasi respon transien sistem.

Kategori	Set Point (rpm)	Time Delay (s)	Rise Time (s)	Peak Time (s)	Over-shoot (%)	Settling Time (s)
Tanpa Beban	200	3,8	5,4	9,3	64,5	42
	400	5,9	8,7	11,1	26,5	30
	600	7,2	10,9	17,5	65	55
	800	9,1	16,8	18,6	10,5	30
Berbeban	200	5,2	7,1	13,3	62,5	43
	400	8	12	15	21,25	43
	600	9,6	15,8	20,2	15	35
	800	11	19,9	25,6	14,25	93

Tabel 4.5 menunjukkan bahwa sistem logika fuzzy kestabilan kecepatan mampu mencapai *set point* yang telah ditentukan. Pada keadaan tanpa beban *settling time* terbaik sistem terjadi pada *set point* 400 rpm dan 800 rpm, sebesar 30 detik. Pada keadaan berbeban *settling time* terbaik sistem terjadi pada *set point* 600 rpm, sebesar 35 detik.

4.6. Pengujian Keandalan Alat Sentrifugasi

Pada pengujian ini mengambil data hasil pemisahan suspensi santan dengan menggunakan alat sentrifugasi hasil perancangan.

Setiap botol pada alat sentrifugasi akan diisi suspensi santan seberat 50 gram. Pengujian akan dilakukan pada beberapa tingkat konsentrasi suspensi santan. Hasil pengujian akan dibandingkan dengan suspensi santan yang telah tersentrifugasi yang sudah disiapkan sebelumnya. Proses sentrifugasi akan berjalan selama 5 menit.



Gambar 4.13 Kondisi santan referensi



Gambar 4.14 Referensi dan bahan uji sebelum sentrifugasi.

Tabel 4.6 Spesifikasi santan referensi.

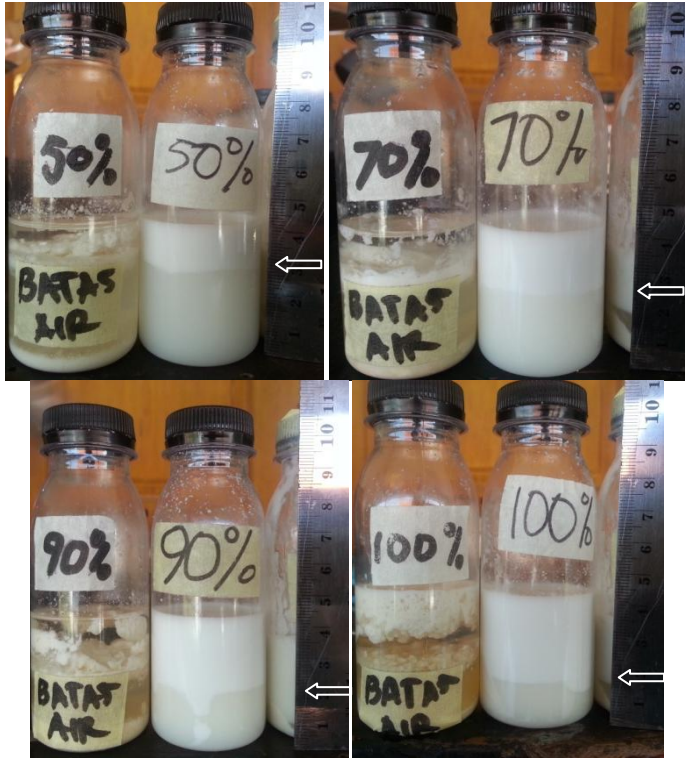
Konsentrasi Santan (%)	Tinggi Air (cm)	Tinggi Santan (cm)	Konsentrasi Air Pada Santan (%)
50	3	4,27	70,25761124
70	2,65	4,28	61,91588785
90	2,4	4,35	55,17241379
100	2	4,5	44,44444444

Tabel 4.7 Spesifikasi pemisahan santan hasil pengujian alat.

Konsentrasi Santan (%)	Tinggi Air (cm)	Tinggi Suspensi (cm)	Konsentrasi Air Pada Santan (%)
50	2,9	4,3	67,44186047
70	2,4	4,3	55,81395349
90	1,9	4,3	44,18604651
100	1,6	4,35	36,7816092

Tabel 4.8 Prosentase keberhasilan.

Konsentrasi Santan (%)	Prosentase Keberhasilan Alat (%)
50	95,99224806
70	90,14480035
90	80,0872093
100	82,75862069
Rata-rata	87,2457196



Gambar 4.15 Kondisi santan setelah disentrifugasi.

Tabel 4.7 menunjukkan bahwa sistem pengendali kecepatan berhasil meningkatkan tingkat keandalan alat sentrifugasi. Dengan rata-rata keberhasilan mencapai 87,24% untuk semua tingkat konsentrasi santan pada pengujian ini. Nilai keberhasilan terbaik terjadi pada level konsentrasi 50% yang mencapai 95,99%, dan nilai terendah terjadi pada level konsentrasi 90% dengan nilai 80,08%.

.....*Halaman ini sengaja dikosongkan*.....

BAB V

PENUTUP

5.1. Kesimpulan

Pada penelitian ini telah dirancang dan dibuat pengendali kecepatan pada alat sentrifugasi alat sentrifugasi menggunakan logika fuzzy. Sistem disusun dari sensor *increment encoder* dan Arduino Uno. Penentuan besar *set point* kecepatan sentrifugasi didasarkan dari nilai masukan, yaitu: konsentrasi santan dan waktu proses. *Set point* kecepatan sentrifugasi akan dijadikan parameter oleh sistem kestabilan kecepatan. Sensor kecepatan yang terdapat pada alat sentrifugasi ini memiliki tingkat *error* ketelitian mencapai 1,52 % pada *duty cycle* PWM sebesar 10 %, dan memiliki kecenderungan *error* semakin besar ketika *duty cycle* dinaikkan. Logika fuzzy penentu *set point* kecepatan sentrifugasi juga mampu menentukan kecepatan alat sentrifugasi berdasarkan perubahan waktu dan konsentrasi santan. Kontroler Logika fuzzy kestabilan kecepatan sentrifugasi juga mampu mengendalikan kecepatan sentrifugasi sesuai dengan kecepatan *set point*. Pada kondisi tanpa beban *settling time* terbaik dari kontroler Logika Fuzzy Kestabilan Kecepatan terjadi pada *set point* 400 dan 800 rpm, dengan waktu 30 detik, dan pada kondisi berbeban *settling time* terbaik terjadi pada *set point* 600 rpm, dengan waktu 30 detik. Alat sentrifugasi juga berhasil meningkatkan kehandalan alat sentrifugasi, terbukti dalam durasi 5 menit berhasil memisahkan santan dengan tingkat keberhasilan rata-rata 87,24 %, dan yang terbaik pada konsentasi 50%, sebesar 95,99%.

5.2. Saran

Dari hasil realisasi Alat Sentrifugasi pada Tugas Akhir ini, terdapat beberapa hal yang masih bisa dikembangkan untuk penelitian selanjutnya, Penambahan sensor suspensi yang berbasis cahaya atau visual. Dibutuhkan sensor suspensi yang mampu mengetahui kondisi suspensi secara *real-time*, sensor berbasis cahaya atau visual karena sensor bisa ditempatkan secara terpisah dari suspensi yang diputar. Pengembangan tidak hanya pada sistem Alat Sentrifugasi, juga perlu pengembangan pada mekanik dari alat sentrifugasi, terutama tempat untuk peletakan suspensi.

.....*Halaman ini sengaja dikosongkan*.....

DAFTAR PUSTAKA

- [1] Ir. Triwibowo Yuwono, Ph. D. *“Biologi Molekular”*, Erlanga, Jakarta, Bab. 3, 2009
- [2] Sri, Kusumadewi, *“Aplikasi Logika Fuzzy Untuk Pendukung Keputusan”*, Graha Ilmu, Yogyakarta, 2010
- [3] Andrianto Heri, Darmawan Aan, *“Arduino Belajar Cepat Dan Pemrograman”*, Informatika, Bandung, 2015
- [4] Mrs. A. A Thorat, Prof. Suhas Yadav, Prof. S. S. Patil, *“Implementation of Fuzzy Logic System for DC Motor Speed Control using Microcontroller”*, International Journal of Engineering Research and Applications, vol. 3, pp. 950-956, Maret-April, 2013.
- [5] A. D. R. Aljabar, Muhammad Rivai, Suwito, *“Rancang Bangun Omni untuk Mencari Sumber Gas Menggunakan Kontrol Logika Fuzzy Berbasis Mikrokontroler ARM STM-32”*, Jurnal Teknik ITS, vol. 7, pp B 17-22, April, 2018
- [6] H. Khoswanto, Djoko Purwanto, *“Keseimbangan Robot Beroda Dua Menggunakan Metode Logika Fuzzy”*, Laporan tesis jurusan Teknik Elektro, Fakultas Teknologi Industri, Universitas Petra, Surabaya, 2010
- [7] A. Khumaedi, N. Soedjarwanto, A. Trisanto. *“Otomatisasi Pengereman Motor DC Secara Elektris Sebagai Referensi Sistem Keamanan Mobil Listrik”*, Jurnal Rekayasa dan Teknologi Elektro, Universitas Lampung, vol. 8, pp 20-30, Januari, 2014.
- [8] W. A. Putri, M. Ilyas, *“Pembuatan Digital Kompas Sebagai Simulasi Penunjuk Arah Pada Objek Menggunakan Komunikasi I2C Berbasis Mikrokontroler Atmega 8535”*, Institut Teknologi Sepuluh Nopember, 2011
- [9] L. Riadhi, M. Rivai, F. Budiman, *“Sistem Pengaturan Oksigen Terlarut Menggunakan Metode Logika Fuzzy Berbasis Mikrokontroler Teensy Board”*, Jurnal Teknik ITS, vol. 6, pp. F-330 – 334, Surabaya, 2017
- [10] D. A. Rahman, M. Rivai, R. Dikairono, *“Sistem Navigasi Pada Balon Udara Menggunakan GPS dan Kontrol Logika Fuzzy”*, Jurnal Teknik ITS, vol. 5, pp A173-178, Juni, 2011
- [11] D. H. Fandiantoro, M. Rivai, R. Dikairono, *“Sistem Penjejak pada Balon Udara dengan Menggunakan Kamera dan Kotrol Logika Fuzzy”*, Jurnal Teknik ITS, vol. 5, A892-897, 2016

- [12] F. A. Muhammad, M. Rivai, Sumito, “ *Perancangan Sistem Stabilisasi Kamera Tiga Sumbu dengan Metode Kontrol Fuzzy untuk Mobile Surveillance Robot*”, Jurnal Teknik ITS, Surabaya, Januari 2016
- [13] K. Sambodho, D. Purnomo H., M. Rivai, “*Identificatio of Levee Strenght for Early Warning System Usng Fuzzy Logic*”, International Conference on Information, Comunication Technology and System, 2014

LAMPIRAN A

Program Pengujian Duty Cycle PWM Terhadap Sistem Penggerak Sentrifugasi:

```
float value=0;
float rev=0;
int rpm, p;
long h, oldtime, waktu, pwm;

void timer1_init(){
    TCCR1A = 0;
    TCCR1B = 0;
    TCNT1 = 0;

    // Set timer1_counter to the correct value for our interrupt
    interval
    // Set CTC Mode, non-PWM
    TCCR1B |= (1 << WGM12);
    // Set Prescaler
    TCCR1B |= (1 << CS11)|(1 << CS10); // 64
    // Set Frekuensi 10 Hz, = 16M/(25*64)-1
    OCR1A = 24999;
    // Set Output Compare A Match Interrupt Enable
    TIMSK1 |= (1 << OCIE1A);
}

void setup()
{
    Serial.begin(9600);
    noInterrupts(); // disable all interrupts
    pinMode(11, OUTPUT);
    pinMode(12, OUTPUT);
    pinMode(13, OUTPUT);
    oldtime=0;
    rev=0;
    pwm=2500;
    p=0;
    timer1_init();
    interrupts();
    attachInterrupt(0,isr,RISING); //attaching the interrupt
}

void tachometer(){
    detachInterrupt(0); //detaches the interrupt
    waktu=millis()-oldtime; //finds the time
    rpm=(rev/waktu)*2000; //calculates rpm
    oldtime=millis(); //saves the current time
    rev=0;
}
```

```

    attachInterrupt(0,isr,RISING);
}

void ccw(){
    digitalWrite(11, HIGH);
    digitalWrite(12, LOW);
}

void pulsa(){
    if (pwm > 17499){
        pwm=2500;
    }
    else{
        while (TCNT1 < pwm){
            digitalWrite(13, HIGH);
        }
        digitalWrite(13, LOW);
    }
}

void isr() //interrupt service routine
{
    rev++;
}

// interrupt service routine
ISR (TIMER1_COMPA_vect){
    h=h+1;
    if (h<250){}
    else{
        ccw();
        tacho();
        pulsa();
    }
}

void loop(){
    p=p+1;
    if (p==145){
        Serial.print("===== BATAS ");
        Serial.print(pwm);
        Serial.println(" =====");
        pwm=pwm+2500;
        p=0;
    }
    Serial.print(rpm);
    Serial.println(" rpm");
    delay(250);
}

```

LAMPIRAN B

Program Pengujian Logika Fuzzy *Set Point* Kecepatan Sentrifugasi

```
float value=0;
float rev=0;
float pwm;
int waktu_proses, konsentrasi, magnitude;
unsigned long pembilang, penyebut;
long speed_ref;

unsigned int a1[5],b1[5];
unsigned long fuzzy_map1 [25];
int speed_rule [25] = {450, 337, 225, 112, 0,
                      552, 377, 450, 225, 0,
                      900, 552, 450, 225, 0,
                      900, 787, 552, 225, 0,
                      900, 900, 552, 337, 0};

void suspensi_waktu(){
    if (waktu_proses>1800){
        konsentrasi=konsentrasi+10;
        waktu_proses=0;
        Serial.println("=====BATAS=====");
        if (konsentrasi>100){
            konsentrasi=40;
        }
    }
}

// Metode Pembentukan Himpunan Fuzzy, Metode Maximum
void fuzzifikasi1 (){
    suspensi_waktu();
    // waktu_proses
    a1[0]=fs_trapesium1(waktu_proses,0,180,540);
    a1[1]=fs_segitiga(waktu_proses,180,540,900);
    a1[2]=fs_segitiga(waktu_proses,540,900,1260);
    a1[3]=fs_segitiga(waktu_proses,900,1260,1800);
    a1[4]=fs_trapesium2(waktu_proses,1260,1620,1800);
    //konsentrasi
    b1[0]=fs_trapesium1(konsentrasi,10,20,30);
    b1[1]=fs_segitiga(konsentrasi,10,30,50);
    b1[2]=fs_segitiga(konsentrasi,30,55,80);
    b1[3]=fs_segitiga(konsentrasi,55,80,100);
    b1[4]=fs_trapesium2(konsentrasi,80,90,100);
}

void minmaks1 (){
    fuzzifikasi1();
```

```

int i=0;
int k=0;
while (i<5){
    int j=0;
    while (j<5){
        fuzzy_map1[k]=0;
        if ((a1[j]!=0)&&(b1[i]!=0)){
            if (a1[j]<=b1[i]){
                fuzzy_map1[k] = a1[j];
            }
            else{
                fuzzy_map1[k]= b1[i];
            }
        }
        j=j+1;
        k=k+1;
    }
    i=i+1;
}
speed_ref = centroid1();
}

// metode defuzzifikasi, metode centroid
long centroid1(){
    int i=0;
    unsigned long z=0;
    unsigned long zu=0;
    while(i<25){
        zu=zu+(fuzzy_map1[i]*speed_rule[i]);
        z=z+fuzzy_map1[i];
        i=i+1;
    }
    pembilang=zu;
    penyebut=z;
    return (zu/z);
}

void setup (){
    Serial.begin(9600);
    waktu_proses=0;
    konsentrasi=40;
}

void loop(){
    minmaks1();
    Serial.print("Konsentrasi = ");
    Serial.println(konsentrasi);
    Serial.print("Detik ke-");
    Serial.println(waktu_proses);
    Serial.print("Kecepatan Referensi = ");

```

```
Serial.println(speed_ref);  
waktu_proses=waktu_proses+15;  
delay(100);  
}
```


LAMPIRAN C

Program Pengujian Logika Fuzzy KestabilanKecepatan Sentrifugasi:

```
float value=0;
float rev=0;
float pwm;
int rpm, oldtime, waktu;
long Vref, error, errorOld, deltaError, high_time, h;

void timer1_init(){
    TCCR1A = 0;
    TCCR1B = 0;
    TCNT1 = 0;

// Set timer1_counter to the correct value for our interrupt
interval
    // Set CTC Mode, non-PWM
    TCCR1B |= (1 << WGM12);
    // Set Prescaler
    TCCR1B |= (1 << CS11)|(1 << CS10); // 64
    // Set Frekuensi 10 Hz, = 16M/(25*)-1
    OCR1A = 24999;
    // Set Output Compare A Match Interrupt Enable
    TIMSK1 |= (1 << OCIE1A);
}

void duty_cycle(){
    fuzzifikasiError();
    minmaks();
    pwm = pwm + (high_time);
    if (pwm>=20000){
        pwm=20000;
    }
    if (pwm<=0){
        pwm=0;
    }
}

void pulsa(){
    while (TCNT1 < pwm){
        digitalWrite(13, HIGH);
    }
}
```

```

        digitalWrite(13, LOW);
    }

    int E[5], dE[5];
    long fuzzy_mapE[25];
    int pwm_rule[25]= {-30, -30, -10,  0, 15,
                       -30, -15,  0, 10, 30,
                       -30, -15,  0, 15, 30,
                       -30, -10,  0, 15, 30,
                       -15,  0, 10, 30, 30};

    int fs_segitiga (int val, int miring_naik, int tengah, int
    miring_turun){
        if ((val >= miring_naik)&&(val<=tengah)) {
            return ((val - miring_naik)*100)/(tengah - miring_naik);
        }
        else if ((val > tengah)&&(val <= miring_turun)) {
            return ((miring_turun - val)*100/(miring_turun - tengah));
        }
        else {
            return 0;
        }
    }

    int fs_trapesium1 (int val, int konstan_awal, int
    konstan_akhir, int miring_turun){
        if (val<=konstan_awal){
            return 100;
        }
        else if ((val >= konstan_awal) && (val <= konstan_akhir)){
            return 100;
        }
        else if ((val >= konstan_akhir)&&(val <= miring_turun)){
            return ((miring_turun-val)*100/(miring_turun-
konstan_akhir));
        }
        else {
            return 0;
        }
    }

    int fs_trapesium2 (int val, int miring_naik, int
    konstan_awal, int konstan_akhir) {
        if ((val >= miring_naik)&&(val <= konstan_awal)){

```



```

    return ((val - miring_naik)*100)/(konstan_awal -
miring_naik);
}
else if ((val >= konstan_awal)&&(val <= konstan_akhir)){
return 100;
}
else if (val >= konstan_akhir){
return 100;
}
else{
return 0;
}
}

void fuzzifikasiError(){
error = Vref - rpm;
deltaError = error - errorOld;
//error
E[0]=fs_trapesium1 (error, -60, -40, -20);
E[1]=fs_segitiga (error, -40, -20, 0);
E[2]=fs_segitiga (error, -20, 0, 20);
E[3]=fs_segitiga (error, 0, 20, 40);
E[4]=fs_trapesium2 (error, 20, 40, 60);
//deltaError
dE[0]=fs_trapesium1 (deltaError, -15, -10, -5);
dE[1]=fs_segitiga (deltaError, -10, -5, 0);
dE[2]=fs_segitiga (deltaError, -5, 0, 5);
dE[3]=fs_segitiga (deltaError, 0, 5, 10);
dE[4]=fs_trapesium2 (deltaError, 5, 10, 15);
//
errorOld = error;
}

void minmaks(){
//fuzzifikasiError();
int i=0;
int k=0;
while(i<5){
int j=0;
while(j<5){
fuzzy_mapE[k]=0;
if ((E[j]!=0) && (dE[i]!=0)){
if (E[j] <= dE[i]){
fuzzy_mapE[k] = E[j];}

```

```

        else{
            fuzzy_mapE[k]= dE[i];
        }
    }
    k=k+1;
    j=j+1;
}
i=i+1;
}
high_time = CoA();
}

long CoA(){
int i=0;
long z=0;
long zu=0;
while (i<25){
    zu = zu + (fuzzy_mapE[i]*pwm_rule[i]);
    z = z + fuzzy_mapE[i];
    i=i+1;
}
return(zu/z);
}

void setup()
{
    Serial.begin(9600);
    noInterrupts();                // disable all interrupts
    pinMode(11, OUTPUT);
    pinMode(12, OUTPUT);
    pinMode(13, OUTPUT);
    oldtime=0;
    rev=0;
    pwm=0;
    Vref=800;
    timer1_init();
    interrupts();                  // enable all interrupts
    attachInterrupt(0,isr,RISING); //attaching the interrupt
}

void tacho(){
    detachInterrupt(0);            //detaches the interrupt
    waktu=millis()-oldtime;        //finds the time
    rpm=(rev/waktu)*2000;          //calculates rpm
}

```

```

    oldtime=millis();           //saves the current time
    rev=0;
    attachInterrupt(0,isr,RISING);
}

// interrupt service routine
void isr(){
    rev++;
}

void ccw(){
    digitalWrite(11, HIGH);
    digitalWrite(12, LOW);
}

// interrupt service routine
ISR(TIMER1_COMPA_vect){
    ccw();
    tacho();
    duty_cycle();
    pulsa();
}

void loop(){
    Serial.println(rpm);
    Serial.println(high_time);
    Serial.println(rev);
    delay(500);
}

```


LAMPIRAN D

Program Sistem Pengendali Kecepatan pada Alat Sentrifugasi dengan Metode Logika Fuzzy:

```
#include "i2c.h"
#include "math.h"
#include "stdio.h"
#include "stdlib.h"
#include "rtc_ds1307.h"
#include "i2c_lcd.h"

int detik,menit,jam,rpm_motor;
int pewaktu;

char buff[33];

int velocity_decision,pulse_decision,pulse_width;

int massa_suspensi,massa_air,konsentrasi_input;

unsigned int miu_t[5],miu_m[5];
unsigned int miu_e[5],miu_de[5];
unsigned int fuzzy_map[25];
int fuzzy_map1[25];

int velocity_rule[25]={65,97,110,110,110
                      ,48,82,97,100,110
                      ,32,65,65,82,97
                      ,15,32,32,32,47
                      ,0,0,0,0,0};

int controller_rule[25]={-30,-15,-5,15,30
                        ,-10,-8,-1,8,10
                        ,-15,-6,0,6,15
                        ,-8,-7,1,7,8
                        ,-8,-2,3,2,8};

//mav filter variable
long int mav_filter[20];
long int mav_out;

//kontroller
float errorP,errorD,error_1;

//pulse rotary
int rotary_pulse;

void fuzzifikasi(){
```

```

//waktu
miu_t[0]= fs_trapesium_ratakiri(pewaktu,0,180,270);
miu_t[1]= fs_segitiga(pewaktu,0,270,1080);
miu_t[2]= fs_segitiga(pewaktu,270,900,1260);
miu_t[3]= fs_segitiga(pewaktu,720,1260,1800);
miu_t[4]= fs_trapesium_ratakanan(pewaktu,1260,1620,1800);
//konsentrasi
miu_m[0]=
fs_trapesium_ratakiri(konsentrasi_input,0,10,30);
miu_m[1]= fs_segitiga(konsentrasi_input,10,30,50);
miu_m[2]= fs_segitiga(konsentrasi_input,30,50,70);
miu_m[3]= fs_segitiga(konsentrasi_input,50,70,90);
miu_m[4]=
fs_trapesium_ratakanan(konsentrasi_input,70,90,100);
//error
miu_e[0]= fs_trapesium_ratakiri(errorP,-1100,-600,-400);
miu_e[1]= fs_segitiga(errorP,-750,-400,0);
miu_e[2]= fs_segitiga(errorP,-400,0,400);
miu_e[3]= fs_segitiga(errorP,0,400,750);
miu_e[4]= fs_trapesium_ratakanan(errorP,400,600,1100);
//errord
miu_de[0]= fs_trapesium_ratakiri(errorD,-1100,-500,-300);
miu_de[1]= fs_segitiga(errorD,-600,-300,0);
miu_de[2]= fs_segitiga(errorD,-300,0,300);
miu_de[3]= fs_segitiga(errorD,0,300,600);
miu_de[4]= fs_trapesium_ratakanan(errorD,300,600,1100);
}

void mamdani(){
    fuzzifikasi();

    //mamdani untuk pulse decision
    int i=0;
    int k=0;
    while(i<5){
        int j=0;
        while(j<5){
            fuzzy_map1[k]=0;
            if((miu_e[j]!=0)&(miu_de[i]!=0)){
                if(miu_e[j]<miu_de[i]){
                    fuzzy_map1[k]= miu_e[j];
                }
                else{
                    fuzzy_map1[k]= miu_de[i];
                }
            }
            k++;
            j++;
        }
        i++;
    }
}

```

```

    }

void decision_make(){ //menggunakan centroid method
    int i=0;
    long int sigm_uD1=0;
    long int sigm_u1=0;
    long int sigm_uD=0;
    long int sigm_u=0;

    while(i<25){
        sigm_uD1= sigm_uD1+(fuzzy_map1[i]*controller_rule[i]);
        sigm_uD= sigm_uD+(fuzzy_map[i]*velocity_rule[i]);
        sigm_u1= sigm_u1+fuzzy_map1[i];
        sigm_u= sigm_u+fuzzy_map[i];
        i++;
    }
    pulse_decision= (sigm_uD1/sigm_u1);
    velocity_decision= (sigm_uD/sigm_u)*10;
    //velocity_decision= 100;
    pulse_width= pulse_width+ pulse_decision;

    if(pulse_width<0){
        pulse_width=0;
    }
    else if(pulse_width>9999){
        pulse_width=9999;
    }
    else{
        pulse_width=pulse_width;
    }
    Serial.println(mav_out);
    //Serial.print("|");
    //Serial.print(pulse_decision);
    //Serial.print("|");
    //Serial.println(OCR1A);
}

void reset_ds1307(){
    //kembalikan detik, menit dan jam pada RTC jadi 0
    ds1307_ctrreg(0x10);
    set_minute(0);
    set_hour(0);
}

int hitung_rpm(int PPR){
    //timer interrupt untuk menghitung rpm
    int temp= rotary_pulse*25*(60/PPR);
    rotary_pulse=0;
    return temp;
}

```

```

}

//initialization

void interrupt1_init(){
    EICRA= EICRA|(1<<ISC11)|(1<<ISC10);
    EIMSK= EIMSK|(1<<INT1);
}
/*
//output PWM
void timer1_init(){
    TCCR1A= 0xA3; //phase correct PWM 10 bit
    TCCR1B= 0x02; //prescaler 64
}
*/

void timer1_init(){
    TCCR1A = 0;
    TCCR1B = 0;
    TCCR1C = 0;
    TCNT1 = 0;

    // set CTC mode
    TCCR1A |= (1 << WGM12);
    // set prescaler 64
    TCCR1B |= (1 << CS11)|(1 << CS10);
    // set frekuensi 25 hz, =16M/(64*25)
    OCR1A = 9999;
    //set Compare match
    TIMSK1 |= (1 << OCIE1A);
}

//timer2 sebagai timer interrupt dengan fs 500Hz
void timer2_init(){
    TCCR2A= (1<<WGM21); //CTC
    TCCR2B= TCCR2B|(1<<CS22)|(1<<CS21)|(1<<CS20); //prescaler
1024
    OCR2A = 255;
    TIMSK2|= (1<<OCIE2A); //interupsi enable timer 2 compare
}

//interrupt sub-routine

ISR(TIMER1_COMPA_vect){
    //get rpm motor
    rpm_motor = hitung_rpm(30);

    //mav_filter
    mav_filter[0]=rpm_motor;
    mav_out=0;
}

```



```

int i=0;
while(i<20){
    mav_out=mav_out+mav_filter[i];
    i++;
}

mav_out= (mav_out/20);

//kontroller PID-Fuzzy
errorP= velocity_decision-mav_out;
errorD= errorP-error_1;
//interrupt untuk fuzzy
mamdani();
if ( TCNT1 < pulse_width ){
    digital.Write(9, HIGH);
}
else{
    digital.Write(9, LOW);}
error_1= errorP;
}

ISR(INT1_vect){
    rotary_pulse++;
}

//another function

char keypad(){
    PORTC= 0x01;
    i2c_address(0x7D);
    int val= i2c_getval(NACK);
    if(val==1){
        i2c_stop();
        return 'C';
    }
    else if(val==2){
        i2c_stop();
        return '0';
    }
    else if(val==4){
        i2c_stop();
        return 'K';
    }
    else{
        i2c_stop();
    }

    PORTC= 0x02;
    i2c_address(0x7D);

```

```

val= i2c_getval(NACK);
if(val==1){
    i2c_stop();
    return '7';
}
else if(val==2){
    i2c_stop();
    return '8';
}
else if(val==4){
    i2c_stop();
    return '9';
}
else{
    i2c_stop();
}

PORTC= 0x04;
i2c_address(0x7D);
val= i2c_getval(NACK);
if(val==1){
    i2c_stop();
    return '4';
}
else if(val==2){
    i2c_stop();
    return '5';
}
else if(val==4){
    i2c_stop();
    return '6';
}
else{
    i2c_stop();
}

PORTC= 0x08;
i2c_address(0x7D);
val= i2c_getval(NACK);
if(val==1){
    i2c_stop();
    return '1';
}
else if(val==2){
    i2c_stop();
    return '2';
}
else if(val==4){
    i2c_stop();
    return '3';
}

```

```

    }
    else{
        i2c_stop();
    }
}

void setup() {
    Serial.begin(9600);

    i2c_init(100);
    i2c_lcdInit();

    DDRC|= 0x0F;
    PORTC|= 0x0F;
    char lala;
    char lala1[10]="";
    int i=0;
    //massa sospensi
    while(lala!='K'){
        i2c_lcdClear();
        i2c_lcdSetCursor(0,0);
        i2c_lcdWriteStr("massa sospensi=");

        i2c_address(0x7C);
        i2c_send(0x00);
        i2c_stop();

        i2c_lcdSetCursor(1,0);
        lala= keypad();

        if(lala=='C'){
            for(int j=0;j<10;j++){
                lala1[j]=0;
                i=0;
            }
        }
        else if((lala>0)&(lala!='C')&(lala!='K')){
            lala1[i]=lala;
            i++;
        }

        i2c_lcdWriteStr(lala1);

        i2c_lcdSetCursor(1,12);
        i2c_lcdWriteStr("gram");

        delay(500);
    }
    massa_suspensi= atoi(lala1);
}

```

```

//massa air
char lolol;
char lolol1[10]="";
i=0;
while(lolol!='K'){
    i2c_lcdClear();
    i2c_lcdSetCursor(0,0);
    i2c_lcdWriteStr("massa air=");

    i2c_address(0x7C);
    i2c_send(0x00);
    i2c_stop();

    i2c_lcdSetCursor(1,0);
    lolol= keypad();

    if(lolol=='C'){
        for(int j=0;j<10;j++){
            lolol1[j]=0;
            i=0;
        }
    }
    else if((lolol>0)&(lolol!='C')&(lolol!='K')){
        lolol1[i]=lolol;
        i++;
    }

    i2c_lcdWriteStr(lolol1);

    i2c_lcdSetCursor(1,12);
    i2c_lcdWriteStr("gram");

    delay(500);
}
massa_air= atoi(lolol1);

konsentrasi_input=
(massa_suspensi*100)/(massa_suspensi+massa_air);

DDRC=0x00;
PORTC=0x00;

DDRB= 0x02; //untuk pinB 1 sebagai output
PORTB= 0x02;

DDRD= 0x00; //untuk pinD 4 sbg T0 input, EXTINT0 input
PORTD= 0x00;

timer1_init();
timer2_init();

```

```

interrupt1_init();
reset_ds1307();

//variable kontroller
errorP=0;
errorD=0;
error_1=0;

//reset rotary pulse
rotary_pulse=0;
pulse_width=0;
}

void loop() {
    // put your main code here, to run repeatedly:
    //get data dari ds1307
    detik= get_seconds();
    menit= get_minute();
    jam= get_hour();
    pewaktu= (jam*3600)+(menit*60)+detik;

    //tampilan-tampilan
    i2c_lcdClear();
    i2c_lcdSetCursor(0,0);
    sprintf(buff,"tmr=%d:%d:%d",jam,menit,detik);
    i2c_lcdWriteStr(buff);

    i2c_lcdSetCursor(0,11);
    sprintf(buff,"|%3d%c",konsentrasi_input,0x25);
    i2c_lcdWriteStr(buff);

    i2c_lcdSetCursor(1,0);
    sprintf(buff,"rpm=%5d|%5d",velocity_decision,mav_out);
    i2c_lcdWriteStr(buff);

    delay(500);
}

```

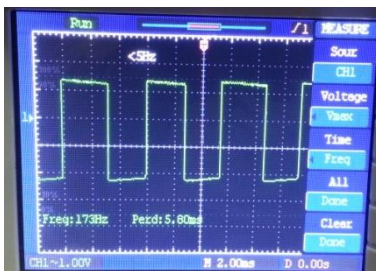

LAMPIRAN E (DOKUMENTASI)

1. Pengujian Kecepatan Maksimal Alat Sentrifugasi

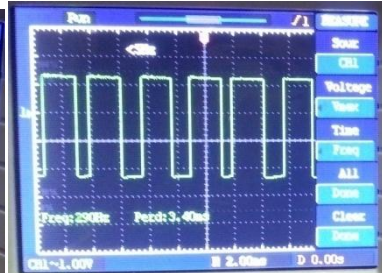


Beban pada pengujian alat.

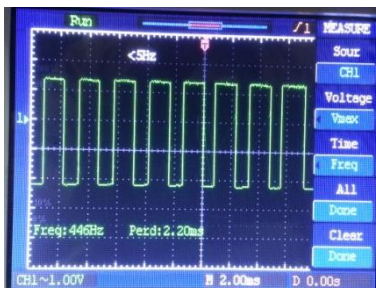
2. Pengujian Sinyal Keluaran Sensor *Increment Encoder*



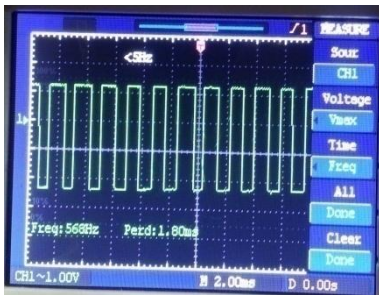
Tegangan 3 Volt



Tegangan 6 Volt



Tegangan 9 Volt Tegangan 12 Volt



Tegangan 15 Volt



Tegangan 18 Volt



Tegangan 3 volt



Tegangan 6 volt



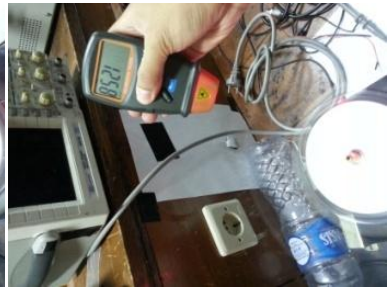
Tegangan 9 volt



Tegangan 12 volt



Tegangan 15 volt



Tegangan 18 volt

BIODATA PENULIS



Lahir 26 tahun lalu di kota kecil, Banyuwangi, 05 Februari 1992 untuk lebih detailnya, yang diperkirakan pada saat itu adalah bulan Sya'ban. Penulis dilahirkan ke dunia dari rahim sang Ibunda. Pada tahun 1997 kedua orang tua penulis memutuskan untuk memulai pendidikan penulis di Sekolah Dasar Negeri 04 Sumbersari yang sangat dekat dari rumah. Lulus SD pada tahun 2004, penulis dimasukkan SLTP Bustanul Makmur Genteng yang sangat jauh dari rumah. Lulus SLTP pada tahun 2007, penulis berhasil masuk SMA Negeri 1 Genteng. Lulus pada tahun 2010 penulis ditakdirkan masuk Jurusan Teknik Elektro, Institut Teknologi Sepuluh Nopember. Menjalani keseharian kuliah-kosan kuliah-kosan, akhirnya dipertengahan tahun 2018 penulis diberikan kelulusan dari Institut Perjuangan ini.